Efficient Topic-based Unsupervised Name Disambiguation

Yang Song[1], Jian Huang[2], Isaac G. Councill[2], Jia Li[3,1], C. Lee Giles[2,1]

[1]Department of Computer Science and Engineering, The Pennsylvania State University University Park, PA 16802, USA

[2]Information Sciences and Technology, The Pennsylvania State University University Park, PA 16802, USA

[3]Department of Statistics, The Pennsylvania State University University Park, PA 16802, USA

ABSTRACT

Name ambiguity is a special case of identity uncertainty where one person can be referenced by multiple name variations in different situations or even share the same name with other people. In this paper, we focus on the problem of disambiguating person names within web pages and scientific documents. We present an efficient and effective two-stage approach. In the first stage, two novel topic-based models are proposed by extending two hierarchical Bayesian text models, namely Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA). Our models explicitly introduce a new variable for persons and learn the distribution of topics with regard to persons and words. After learning an initial model, the topic distributions are treated as feature sets and names are disambiguated by leveraging a hierarchical agglomerative clustering method. Experiments on web data and scientific documents from CiteSeer indicate that our approach consistently outperforms other unsupervised learning methods such as spectral clustering and DBSCAN clustering and could be extended to other research fields. We empirically addressed the issue of scalability by disambiguating authors in over 750,000 papers from the entire CiteSeer dataset.

Keywords

Unsupervised Learning, Bayesian Models, Name Disambiguation, Probability Analysis.

1. INTRODUCTION

With the emergence of major search engines like Google and Yahoo! that automate the process of gathering web pages to facilitate searching, it has become increasingly common for Internet users to search for their desired results to specific queries through search engines, with name queries making up approximately 5-10% of all searchers. Name queries are usually treated by search engines as normal keyword searches without attention to the ambiguity of particular names. For example, searching Google for "Yang Song" results in more than 11,000,000 pages with the same person's name, of which even the first page shows five different people's home pages. Table 1 lists the first four results which correspond to four different people. Due to this heterogeneous nature of data on the Internet crawled by search engines, the issue of identity uncertainty or *name ambiguity* has attracted significant research attention. Beyond the problem of sharing the same name among different people, name misspelling, name abbreviations and other reference variations compound the challenge of name disambiguation.

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

Table 1: First 4 search results of the query "Yang Song" from Google that refer to 4 different people.

The same issue also exists in most Digital Libraries (DL), hampering the performance and quality of information retrieval and credit attribution. In DL such as DBLP[1] and CiteSeer [7], textual information is stored in metadata records to speed up field searching, including titles, venues, author names and other data. However, the existence of both *synonyms* and *polysems* as well as typographical errors makes the problem of disambiguating author names in bibliographies (citations) non-trivial. In the case of *synonyms*, an author may have multiple name variations/abbreviations in citations across publications, e.g., the author "C. Lee Giles" is sometimes used as "C. L. Giles" in his citations. For *polysems*, different authors may share the same name label in multiple citations, e.g., both "Guangyu Chen" and "Guilin Chen" are used as "G. Chen" in their citations. In addi-

JCDL2007, June 17-23, 2007, British Columbia, Canada.

[1]http://www.informatik.uni-trier.de/∼ley/db/index.html

tion to the issue of citations, authors may be inclined to use different name variations even in the title pages of their publications due to a variety of reasons (such as the change of their maiden names).

Existing approaches that address the issue of name disambiguation generally fall into two categories: supervised learning and unsupervised learning methods. In the case of supervised learning [8], the objective is to determine the *name label* by leveraging the related information (e.g., page contents and citation information). Careful labeling with specific domain knowledge is usually required for supervised learning, which makes it both error-prone and label intensive. Comparatively, unsupervised learning methods [9, 1] do not require manual labeling but instead prudently choose features (e.g., social networks, link structures, co-authorship) to classify similar instances into groups or clusters. A variety of clustering methods including K-means and spectral clustering have been extensively utilized and compared for unsupervised name disambiguation. Nevertheless, choosing the right set of features often results in better performance than exhaustively seeking the best clustering method. However, supervised learning methods generally achieve better performance with the trade-off of expensive training time.

1.1 Our Contribution

The objective of this paper is to produce an approach that includes the attractive properties of both supervised and unsupervised learning methods while trying to avoid the respective limitations. Specifically, we explore the use of a two-stage approach to address the problem of disambiguating person names in both web appearances and scientific documents (including citations). During the first stage, we present two novel *topic-based* models inspired by two generative models for documents: Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA). Our models differ from the general methods by explicitly introducing a variable for *persons*. After an initial model is built, person names are disambiguated by leveraging an unsupervised hierarchical agglomerative clustering method [4], which groups similar instances together in a bottom-up fashion. We empirically study our models by comparing against three other clustering methods on both web data and scientific documents.

The underlying rationale for using generative models with latent variables is to harness the unique topic distribution related to different persons. For example, the basketball player "Michael Jordan" is more likely to appear in the topic *sports*, while Professor "Michael Jordan" in Berkeley may have high probability of being associated with the topic *academics*. Likewise, for the authors of scientific papers, one may have his/her own focus, e.g. Professor "Jia Li" in the math department of Alabama and Professor "Jia Li" in the statistics department of Penn State. Moreover, even authors within the same research field should be distinguishable by topics, e.g. two researchers named "Amit Kumar" working separately at Cornell and Rice are both involved in research on networks, but with specific focus on network routing and wireless networks respectively. As a result, topic distribution may be a useful *feature set* that allows us to distinguish people from each other in a principled and efficient way.

Although both PLSA and LDA have been extensively studied and applied to various applications, there has been relatively few comparisons between the their performance in real-world studies except in [3]. Theoretically, PLSA does not need to make any assumptions regarding the document distribution, thus it is more flexible when dealing with abnormal data sets. Meanwhile, the LDA (Bayesian) approach is more robust on sparse data. With a large feature space, LDA generally exhibits better performance than PLSA as well as other probabilistic models.

This paper is organized as follows: Section 2 presents previous work on generative models and name disambiguation; Section 3 and Section 4 propose our extended PLSA and LDA models, respectively; Section 5 discusses the use of agglomerative clustering for name disambiguation; Section 6 probes the advantages of our models through a number of experiments; we conclude with future work in Section 7.

2. RELATED WORK

Generative Models for Documents

Using generative models for characterizing documents as well as images has become a recent trend in machine learning research. The first well-known model was introduced by Deerwester [6], namely *Latent Semantic Analysis* (LSA). The key idea of LSA is to map high-dimensional input data to a lower dimensional representation in a *latent semantic space* that reflects semantic relations between words, the mapping was done by Singular Value Decomposition (SVD), and thus restricted to be linear. LSA assumes that there are K underlying latent topics, to which documents are generated accordingly. Those latent topics are assumed to be approximately the same as document classes, resulting in a significant compression of data in large collections.

From a statistical point of view, Hofmann [10] presented an alternative to LSA, or Probabilistic Latent Semantic Analysis/Indexing (PLSA/PLSI), which discovers sets of latent variables with a more solid statistical foundation. The model is described as an *aspect model* that is essentially a latent class statistical mixture model, assuming the existence of hidden factors underlying the co-occurrences among two sets of objects. Thus, a single word is generated from a single topic while different words may belong to different topics within a document. *Expectation-Maximization* (EM) algorithm is applied for the inference of parameters in this model that maximize the likelihood of the data. An obvious problem of PLSA is that the model has a number of parameters that grow linearly with the size of the document collection, yielding a large potential for overfitting. Due to its efficiency and flexibility, PLSA has been widely used in many research fields, including collaborative filtering [11] and web information retrieval [24, 14].

Blei et al. later introduced a Bayesian hierarchical model, *Latent Dirichlet Allocation* (LDA) [3], in which each document has its own topic distribution, drawn from a conjugate Dirichlet prior that remains the same for all documents in a collection. The words within that document are then generated by choosing a topic from this distribution. A word is picked from that topic according to the posterior probability of the topic, which is determined by another Dirichlet prior. Inference of parameters and model learning are performed efficiently via variational EM algorithm, since exact inference is intractable in LDA due to the coupling of parameters. Essentially, this model can be statistically treated as a fully generative aspect model, which assumes an exchangeability for words and topics in documents. Experimental results indicate that LDA has better generalization

performance than PLSA and a mixture of unigrams model as well as higher classification accuracies and better predictions of user preferences in the task of collaborative filtering. Successful applications and extensions of the LDA model includes unsupervised document retrieval [22] and time series analysis [21].

Name Disambiguation

Prior name disambiguation research can be categorized into supervised classification and unsupervised clustering. In [8], different classification methods such as hybrid Naive Bayes and Support Vector Machines (SVM) have been applied to a DBLP dataset. In large-scale digital libraries, however, supervised classification is inappropriate due to the unaffordable cost of human annotation for each name.

Different clustering methods have also been applied in the literature. Earlier approaches such as hierarchical clustering [18] suffered from the transitivity problem[2]. Han et al [9] used a more sophisticated K-spectral clustering method to cluster author appearances. While Han's method could find an approximation of the global optimal solution (in terms of a criteria function) for a sampled dataset, it is unsuitable for large-scale digital libraries since K is not known a priori for an ever increasing digital library and the computational complexity $O(N^2)$ is intractable for N=739,135 in CiteSeer. Lee et al. [16] successfully addressed the scalability issue by using a two level blocking framework; however, this resulted in inconsistent labeling due to the transitivity problem in such a solution. In [12], used a SVM-based distance function was used to calculate the similarity of the metadata records of author appearances, and explicitly solved the transitivity problem in labeling with the DBSCAN clustering method. [2] proposed an LDA-based entity resolution method which is generative and does not require pair-wise decisions.

The aforementioned work mainly tackled the name disambiguation problem using the metadata records of the authors. This paper solves the name disambiguation problem in a novel way, by accounting for the topic distribution of the authors and adopting unsupervised methods. As such it yields an accurate and highly efficient solution to the person name disambiguation problem.

3. TOPIC-BASED PLSA

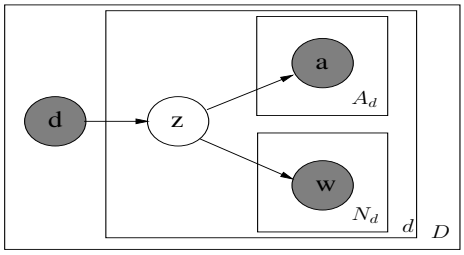
We use the following notations in this paper.

- A *document* d is a sequence of N words denoted by $\mathbf{w} = \{w_1, w_2, ..., w_N\}$, where w_n denotes the nth word in a document, plus a sequence of M name appearances denoted by $\mathbf{a} = \{a_1, a_2, ..., a_M\}$, where a_j represents the jth name appearances in the document. For web data, name appearances refer to the owners of their homepages or the subject of the articles. For scientific documents, it means the authors of the papers as well as the authors in the citations.
- A *corpus* is a collection of T documents denoted by $D = \{d_1, d_2, ..., d_T\}$.
- $W = \{w_1, ..., w_p\}$ represents the number of unique words (i.e., vocabulary) in a corpus with size p. $A = \{a_1, ..., a_q\}$ indicates the number of name appearances in a corpus with size q.

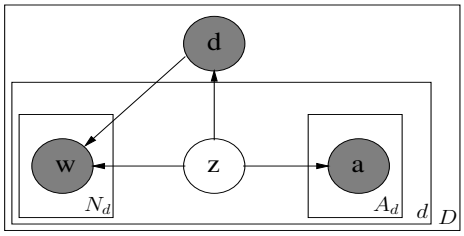
[2]The transitivity problem is a name A is co-referent with B, and B with C, while A is not co-referent with C. C.f. [12] for more detailed discussions.

- The relationships between documents, names and words are connected by a set of latent variables $Z = \{z_1, ..., z_K\}$ with size K, each of which represents a latent topic.

In our document-name-word scenario, an observation is treated as a triplet $\{d, a, w\}$ that represents an instance that a name a appears in document d, which contains the word w. The relationship inherent in the triplets is associated by a set of topics Z. Our mixture model has a conditional independence assumption of variables, i.e., the observed objects are conditionally independent on the state of the related latent variables, which are essentially treated as persons' interests. Specifically, a document d is potentially related to several topics Z with different probabilities, and the latent variables consequently generate a set of words $\mathbf{w}$ and name appearances $\mathbf{a}$ that are closely related to a specific topic. Figure 1 (a) shows the graphical illustration of the generative model.



(a) The three-way aspect model



(b) An alternative view of the aspect model

Figure 1: Graphical model representation. (a) The original document-name-word model, D is the number of documents, N_d is the number of words in document d and A_d is the number of name appearances in document d. (b) The alternative view of the model. Shaded nodes are observed variables.

3.1 The Aspect Model

The joint probability of the aspect model over $d \times a \times w$ is defined as the mixture:

$$P(d, a, w) = P(d)P(a, w|d) \tag{1}$$

$$P(a, w|d) = \sum_{z \in Z} P(a, w|z)P(z|d) \tag{2}$$

The definition of the generative model can be described in the following procedure:

1. pick a document d from the corpus D with probability $P(d)$,
2. select a latent class z_k with probability $P(z_k|d)$,

3. generate a word w with probability $P(w|z_k)$,

4. generate a name a with probability $P(a|z_k)$.

In this model, we introduce a set of latent variables z that breaks the direct relationships between documents, words and names, i.e., they are conditionally independent but still associated through latent variables. Note that by reversing the arrow from documents and words to latent topics, an equivalent symmetric model as shown in Figure 1 (b) can be parameterized by

$$P(d, a, w) = \sum_{z \in Z} P(z)P(d|z)P(w|z)P(a|z). \quad (3)$$

This paper will focus on Figure 1 (a) for inference unless otherwise mentioned.

3.2 Model Fitting with the EM Algorithm

The goal of model fitting for PLSA is to estimate the parameters $P(z), P(a|z), P(z|d), P(w|z)$, given a set of observations (d, a, w). The standard way to estimate the probability values is the Expectation-Maximization (EM) algorithm [5], which alternates two steps: (1) an expectation (E) step where posterior probabilities are estimated for the latent variables, based on the current estimates of the parameters; and (2) a maximization (M) step where parameters are estimated again to maximize the expectation of the complete data (log) likelihood. In the E-step, we compute

$$P(z|d, a, w) \propto \frac{P(z)P(a|z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(a|z')P(d|z')P(w|z')}. \quad (4)$$

In the M-step, we aim at maximizing the expectation of the complete data likelihood, the formulas are:

$$P(a|z) \propto \frac{\sum_{d,w} n(d, a, w)P(z|d, a, w)}{\sum_{d,a',w} n(d, a', w)P(z|d, a', w)} \quad (5)$$

$$P(w|z) \propto \frac{\sum_{a,d} n(d, a, w)P(z|d, a, w)}{\sum_{d,a,w'} n(d, a, w')P(z|d, a, w')} \quad (6)$$

$$P(z|d) \propto \frac{\sum_{a,w} n(d, a, w)P(z|d, a, w)}{\sum_{d',a,w} n(d', a, w)P(z|d', a, w)} \quad (7)$$

where $n(d, a, w)$ denotes the number of occurrences of word w in document d with name a. The EM algorithm stops on convergence, i.e., when the improvement of the log-likelihood is significantly small:

$$\mathcal{L} = \sum_{a=1}^{A} \sum_{d=1}^{D} \sum_{w=1}^{W} n(d, a, w) \log P(d, a, w) \quad (8)$$

3.3 Predicting New Name Appearances

Despite the effectiveness of PLSA for mapping the same document to several different topics, it is still not a fully generative model at the level of documents, i.e., the number of parameters that need to be estimated grows proportionally with the size of the training set. Additionally, there is no natural way to assign probability to new documents. Therefore, to predict the topics of new documents (with potentially new names) after training, the estimated $P(w|z)$ parameters are used to estimate $P(a|z)$ for new names a in test document d through a "folding-in" process [10]. Specifically, the E-step is the same as equation (4); however, the M-step maintains the original $P(w|z)$ and only updates $P(a|z)$ as well as $P(z|d)$.

3.4 Probabilistic Inference

The PLSA model mentioned in the above section not only can derive relationships between documents, words and names, but by using probabilistic inference, it can also be used to model the topic patterns for names. Specifically, given $P(a|z)$ the probability of observing a name appearance given a certain topic, we can model the probability that a certain topic is of interest to a given name by simply applying the Bayes rule:

$$P(z|a) \propto \frac{P(a|z)P(z)}{\sum_z P(a|z)P(z)}. \quad (9)$$

In this way people that share similar topics can be modeled through the same pattern. By applying unsupervised learning methods, we can further cluster names for the task of name disambiguation.

4. TOPIC-BASED LDA

In this section, we propose another topic-based Bayesian model. Our model is primarily an extension of the Latent Dirichlet Allocation (LDA) model proposed by Blei et al. in 2003 [3], which has quickly become regarded as one of the most efficient and effective probabilistic modeling algorithm in statistical machine learning.

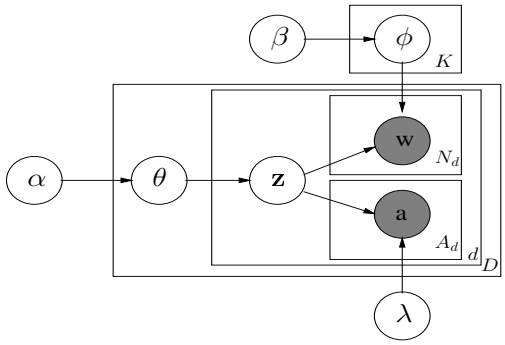
The major difference between PLSA and LDA is that in PLSA the latent variables are dependent on each document, while in LDA the topic mixture is drawn from a conjugate Dirichlet prior which remains the same for all documents. Thus LDA is able to overcome the over-fitting problem in PLSA while naturally generating new documents with consistent generative semantics.

The generative process of our topic-based LDA model can be formalized as follows:

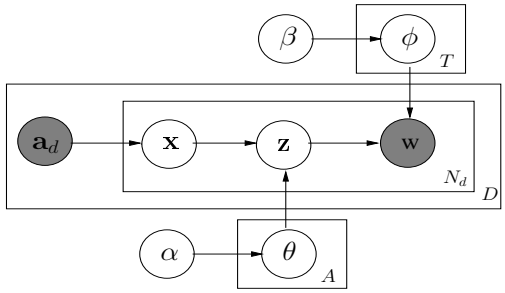
- Draw a multinomial distribution ϕ_z for each topic z from a Dirichlet distribution with prior β;
- For each document d, draw a multinomial distribution θ_d from a Dirichlet distribution with prior α;
- For each word w_{di} in document d, draw a topic z_{di} from the multinomial distribution θ_d;
- Draw a word w_{di} from the multinomial distribution $\phi_{z_{di}}$;
- Draw a name a_{di} from the multinomial distribution $\lambda_{z_{di}}$.

Figure 2 (a) depicts our model. Regarding the generation of parameters, α and β are corpus-level parameters and only sampled once when creating the generative corpus; θ_d are document-level variables, sampled once for each document; z_{di}, w_{di} and a_{di} are word-level variables and need to be sampled once for each word/name in the document.

Although there is resemblance between our proposed LDA model and the author-topic model [19], there exists important differences in the relationship between name appearances and words. In the author-topic model, x denotes an author who is responsible for a given word. In our model, however, names (authors) and words are not directly related, i.e., each topic can generate a set of names and a set of words simultaneously with different probabilities, allowing more freedom to the model in parameter estimation.



(a) Our proposed topic-based LDA model.



(b) The author-topic model [19].

Figure 2: Graphical model representation of the LDA model. (a) Our topic-based model. (b). The author-topic model. K is the number of topics, D is the total number of documents, N_d is the number of tokens in document d and A_d represents the number of name appearances in document d.

4.1 Inference and Parameter Estimation

4.1.1 Inference

The inference problem in LDA is to compute the posterior of the (document-level) hidden variables given a document $d = (\mathbf{w}, \mathbf{a})$ with parameters α and β, i.e., $p(\theta, \phi, \mathbf{z}|\mathbf{w}, \mathbf{a}, \alpha, \beta, \lambda)$,

$$p(\theta, \phi, \mathbf{z}|\mathbf{w}, \mathbf{a}, \alpha, \beta, \lambda) = \frac{p(\theta, \phi, \mathbf{z}, \mathbf{w}, \mathbf{a}|\alpha, \beta, \lambda)}{p(\mathbf{w}, \mathbf{a}|\alpha, \beta, \lambda)}. \quad (10)$$

Here $p(\mathbf{w}, \mathbf{a}|\alpha, \beta, \lambda)$ is usually referred to as the marginal distribution of document d:

$$\begin{aligned} & p(\mathbf{w}, \mathbf{a}|\alpha, \beta, \lambda) \\ = & \iint p(\theta|\alpha)p(\phi|\beta) \prod_{n=1}^{N} p(w_n|\theta, \phi) \prod_{m=1}^{M} p(a_m|\theta, \lambda)\, d\theta d\phi \\ = & \iint p(\theta|\alpha)p(\phi|\beta) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta)p(w_n|z_n, \phi)) \right) \\ & \cdot \left(\prod_{m=1}^{M} \sum_{z_n} p(z_n|\theta)p(a_m|z_n, \lambda)) \right) d\theta d\phi \quad (11) \end{aligned}$$

By marginalizing over the hidden variable z, the name distribution $p(a|\theta, \lambda)$ can be represented as follows:

$$p(a|\theta, \lambda) = \sum_{z} p(a|z, \lambda)p(z|\theta) \quad (12)$$

As a result, the likelihood of a document collection D could be calculated by taking the product of the marginal probabilities of individual documents,

$$\begin{aligned} & p(D|\alpha, \beta, \lambda) = \\ & \iint \prod_{z=1}^{K} p(\phi_z|\beta) \prod_{d=1}^{N} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(w_n|\theta, \phi) \right) \\ & \cdot \left(\prod_{m=1}^{M} p(a_m|\theta, \lambda) \right) d\theta d\phi \quad (13) \end{aligned}$$

Unfortunately, inference cannot be performed exactly on this model due to the problematic coupling between parameters θ, ϕ and λ. Alternative methods have been widely developed to approximate the inference, including variational inference [3] and other methods. In the following section, we apply the Gibbs sampling framework to get around the intractability problem of parameter estimation.

4.1.2 Gibbs sampling for the LDA model

The Gibbs sampling algorithm was developed as a special case of the Markov Chain Monte Carlo (MCMC) algorithm, which estimates the complex joint probability distribution of several variables by generating random samples from the observed data. Note that the sampling algorithm is actually used to derive conditional probabilities for the sampler. Specifically, we need to know the conditional probabilities $p(\theta_m|\alpha, z_{m1}, ..., z_{mN})$, $p(z_{mn}|\theta_m, w_{mn}, \beta)$, where $m = 1, ..., M$ and $n = 1, ..., N$.

We construct a Markov chain that converges to the posterior distribution on $\mathbf{z}$ and then use the results to infer θ and ϕ, i.e., $p(\mathbf{z}|\mathbf{w}, \mathbf{a})$.

Based on the graphical representation in Figure 2, the posterior distribution can be derived as follows:

$$p(z_i = j|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{a}) \propto p(z_i = j|\mathbf{z}_{-i})p(w_i|\mathbf{z}, \mathbf{w}_{-i})p(a_i|\mathbf{z}, \mathbf{a}_{-i}) \quad (14)$$

$$\propto \frac{H_{dj}^{DT} + \alpha}{\sum_{j'} H_{dj'}^{DT} + K\alpha} \frac{H_{mj}^{WT} + \beta}{\sum_{m'} H_{m'j}^{WT} + W\beta}, \quad (15)$$

Notations	Explanations
W	number of words (vocabulary)
K	number of topics
D	number of documents
A	number of name appearances
$z_i = j$	the assignment of the ith word in a document to topic j
$\mathbf{z}_{-i}$	all topic assignments not including the ith word, i.e., $\{z_1, ..., z_{i-1}, z_{i+1}, ..., z_K\}$
H_{mj}^{WT}	number of times word m assigned to topic j, except the current instance
H_{dj}^{DT}	number of times document d contains topic j, except the current instance
ϕ_{mj}	the probability of using word m in topic j
θ_{dj}	the probability of document d contains topic j

Table 2: Notations used for Gibbs sampling.

where the first two terms of Equation (15) is inferred by following the Dirichlet distribution derivation.

Note that in our case, we do not estimate the parameters α, β and λ. For simplicity and performance, they are fixed at $50/K$, 0.01 and 0.1 respectively.

Equation (13) is considered as the conditional probability of the random variables θ and ϕ. For any individual sample, we can estimate them from the latent variable $\mathbf{z}$ by

$$\hat{\theta}_{dj} = \frac{H_{dj}^{DT} + \alpha}{\sum_{j'} H_{dj'}^{DT} + K\alpha} \tag{16}$$

$$\hat{\phi}_{mj} = \frac{H_{mj}^{WT} + \beta}{\sum_{m'} H_{m'j}^{WT} + W\beta} \tag{17}$$

5. PEOPLE NAME DISAMBIGUATION

Learning both the PLSA and LDA models is equivalent to learning the probability distribution of the topic-word $P(w|z)$ and the topic-name $P(a|z)$ matrices. However, the topic-name matrix only reflects the relationships between names and topics, thus several people may have very similar topic interests, especially those from the same research group. For the purpose of name disambiguation, the topic-name matrix is processed further with a hierarchical clustering method. We extend the original agglomerative clustering method for our task, since it has been shown that the bottom-up clustering method performs better than the K-means method as well as other top-down clustering methods in terms of both computational cost and clustering accuracy, particularly when the number of desired clusters is not significantly smaller than the number of points.

5.1 Agglomerative Clustering

To distinguish people that have similar topic interests but with different names, we generate a name-name matrix that measures the pairwise similarity between names. Levenshtein distance [17] (defined as $Le(x, y)$) is used as the measurement and as a result the similarity between two names x and y can be defined as follows ($|\cdot|$ represents the length of the string):

$$Sim(x, y) = 1 - \frac{Le(x, y)}{max(|x|, |y|)}. \tag{18}$$

Our modified agglomerative clustering method is shown in Algorithm 1, in which each name a_i is a vector of length K, a_{ik} reflects the probabilities of name a_i being in a specific topic k, and satisfies $\sum_k a_{ik} = 1$. We apply Euclidean distance as our point-level distance metric, i.e., $\mathcal{D}(a_i, a_j) = \sqrt{\sum_k (a_{ik} - a_{jk})^2}$. Meanwhile, to measure the distance between clusters, the complete-link metric [15] is used that considers the maximum distance of all elements in two clusters[3]. Two additional parameters should also be specified at the beginning of the algorithm, ϵ and θ, as the stopping criteria for the entire program and the merge criteria for two names/name clusters, respectively. In practice, we set $\epsilon = 0.05$ and $\theta = 0.5$.

6. EXPERIMENTS

[3]We also tried both single-link algorithm [20] and wards method [13], the performance are almost equally well.

Algorithm 1 Agglomerative Clustering

1: **Input:**
   $a_1, ...a_M$: names to cluster
   $\mathcal{D}(a_i, a_j)$: point-level distance metric
   $\mathcal{C}(c_i, c_j)$: cluster-level distance metric
   $Sim(a_i, a_j)$: name-name similarity matrix
   ϵ, θ: threshold parameter
2: **Initialize**
   place each name in a singleton cluster,
   calculate the pairwise distance between names according to $\mathcal{D}$,
   set $\mathcal{C} \leftarrow \mathcal{D}$,
3: **Clustering Procedure**
4: **Repeat**
   find two names (a_i, a_j) or name clusters (c_i, c_j) that are closest according to $\mathcal{D}$ and $\mathcal{C}$,
   randomly choose a name to represent a cluster,
   if $Sim(a_i, a_j)$ is greater than θ
      merge the pair to form a new cluster,
   else
      find the next closest pair or quit if no pair satisfy the criteria,
   update the distance between clusters according to $\mathcal{C}$,
5: **Until** the distance between the closest pair of any two clusters is greater than ϵ,
6: **Output:** Clusters $c_1, ...c_\tau$.

To evaluate the two proposed methods, we perform the experiments on two applications, i.e., disambiguation of people's web appearances and author names in scientific documents.

6.1 Evaluation Metrics

Instead of using a matching matrix (a.k.a. a confusion matrix in supervised learning) as in [9] (since the number of clusters K needs to be specified explicitly in advance, making it inappropriate for unsupervised learning), two sets of metrics are applied in our experiments as in [23, 12], namely **pair-level pairwise F1** score $F1P$ and **cluster-level pairwise F1** score $F1C$. $F1P$ is defined as the *harmonic mean*[4] of **pairwise precision** *pp* and **pairwise recall** *pr*, where *pp* is measured by the fraction of co-referent pairs in the same cluster, and *pr* the fraction of co-referent pairs placed in the same cluster. Likewise, $F1C$ is the harmonic mean of **cluster precision** *cp* and **cluster recall** *cr*, where *cp* is the fraction of totally correct clusters to the number of clusters acquired by the algorithm, and *cr* is the fraction of true clusters to that of the algorithm.

As the baseline method, we extracted names from the contents and formed a *name-word* matrix, which was augmented by the standard *tf-idf* method, we then applied the agglomerative clustering using inter-cluster closeness as the measure (Agglo). Our methods are further compared with two unsupervised learning approaches, the k-way spectral clustering (Spectral) [9] and the LASVM+DBSCAN approach (DBSCAN) as described in [12].

The most influential parameter on the performance as well as the scalability of our models is the number of topics K. Following convention [10, 3], we chose the values of K from the set $\{2, 5, 10, 20, 50, 100, 200\}$. For interests of space, only

[4]$H(x_1, x_2) = \frac{2x_1x_2}{x_1+x_2}$

the best results with optimal K are reported. Meanwhile, as mentioned above, the priors α, β and λ for the LDA model are chosen as $50/K$, 0.01 and 0.1 respectively.

6.2 Web Appearances of Person Names

In this section, we consider the problem of automatic disambiguation of person names on the web. To be specific, when users submit name queries like "Michael Jordan" to search engines, we want to distinguish name results by the content of the retrieved web pages. We utilize the public data set[5] generated by Ron Bekkerman and Andrew McCallum [1]. 12 person names including SRI employees and professors (e.g., "David Israel" and "Andrew Ng") are submitted as queries to the Google search engine, the first 100 pages are then retrieved for each query. Post-processing is performed to clean the pages, resulting in a total of 1,085 web pages referring to 187 different people. All pages are manually labeled in the title indicating the position of the person. Among these web pages, 420 are found relevant to the 12 particular names. Some statistics can be found in [1].

For our experiment, the data set is further processed. We first translate the titles into labels with +1 indicating relevant and -1 otherwise. All URLs included in the pages are removed as well as other trivial characters. We then use the rainbow[6] tool to process the remaining text. Stemming and stop words removal are performed, words that appear less than twice are removed as well. Furthermore, to eliminate the bias towards longer documents, only the first 200 words are used in each example.

Table 3 summarizes the clustering results regarding the F1P and F1C scores. Overall, our topic-based models consistently outperform other methods for both metrics, with more than 90% on F1P score and 75% on F1C score on average. For most of the people, both PLSA and LDA achieve the best performance with 10 topics, which decrease sharply with the increase of topic numbers. The highest F1P scores for both models are achieved from the class "Leslie Pack Kaelbling", since it only has two namesakes in that class. For the "Tom Mitchell" class that has 37 namesakes, our methods are still able to achieve 85% and 82.4% F1P scores respectively, with the trade-off of using more topics (20) to disambiguate. Generally, the performance decreases and the number of topics increases with more namesakes in the class. Regarding the cluster F1 scores, since no credits will be given to clusters that are *partially* correct (i.e., either having more or less instances than the real clusters), the performance is commonly worse than the pair-wise metrics. The best F1C scores are achieved in the class "Andrew Ng" which has 29 namesakes, larger number of topics (50 and 20 for PLSA and LDA respectively) shows better performance.

Figure 3 plots the result of the McCallum class for both models by projecting the data matrix on the first three eigenvectors. We choose two clusters for visualization here, one is "Andrew McCallum" from UMass and other people with the identical name for the other cluster. It is evident that both models have very high clustering accuracies and separate two clusters quite well. Specifically, PLSA only misclassified one positive instance to be negative while LDA misclassified one negative instance to be positive.

[5]http://www.cs.umass.edu/∼ronb

[6]http://www.cs.cmu.edu/∼mccallum/bow

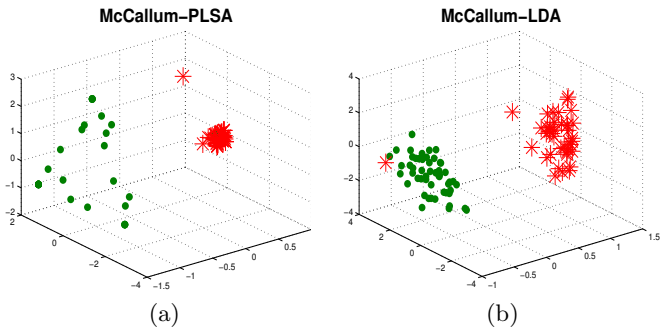


Figure 3: 3D visualization of feature distribution of the *name-topic* matrix in the web appearances data set. *'s indicate the positive class (i.e. "Andrew McCallum" from UMass) and · represents negative classes. (a). PLSA result. (b). LDA result.

Name	Variations	Records
A. Gupta	44	506
A. Kumar	36	143
C. Chen	103	536
D. Johnson	41	350
J. Robinson	30	115
J. Smith	86	743
K. Tanaka	20	53
M. Jones	53	352
M. Miller	34	230
Mean	49.7	336.4

Table 4: Summary of the 9 CiteSeer data sets of different author names and the data size. These names are most representative for the worst case scenario in author name appearances in scientific documents.

6.3 Author Appearances in Scientific Documents

To disambiguate author appearances in the scientific documents, we collect data from the CiteSeer Digital Library.

CiteSeer is currently one of the largest digital libraries that holds more than 750,000 documents, primarily in the domain of computer and information science. CiteSeer indexes several kinds of data formats (*txt, PDF, PS*); however, for our experiment, we convert non-text formats into text and only make use of plain text files. For the purpose of efficiency, extraction is performed only from the summarizing parts (title, author names, abstracts and keyword fields) and the first page of each document.

We obtained the nine most ambiguous author names from the entire data set as shown in Table 4, each of which has at least 20 name variations. In the worst case (C. Chen), 103 authors share the same name.

Two steps of pre-processing are performed before the experiments. First, author names are extracted from individual documents, each of which contains the author metadata associated with a unique paper identifier. Second, author references are extracted from citations by regular expressions and manual correction. Rainbow is then applied to form the document-term and document-author matrices.

Figure 4 plots the results of the CiteSeer data set on F1P scores and F1C scores. Clearly, our methods consistently

	Num of pages	Agglo		Spectral		DBSCAN		PLSA+Agglo		LDA+Agglo	
		F1P	F1C	F1P	F1C	F1P	F1C	F1P	F1C	F1P	F1C
Cheyer	97	0.580	0.211	0.602	0.333	0.852	0.650	0.920	0.677 (10)	0.935	0.725 (20)
Cohen	88	0.515	0.208	0.500	0.210	0.742	0.520	0.888	0.625 (10)	0.850	0.625 (10)
Hardt	81	0.350	0.159	0.362	0.267	0.744	0.577	0.755	0.625 (5)	0.875	0.717 (10)
Israel	92	0.700	0.455	0.720	0.466	0.855	0.680	0.952	0.877 (20)	0.975	0.841 (20)
Kaelbling	89	0.825	0.425	0.825	0.425	0.875	0.739	0.972	0.757 (10)	0.955	0.767 (20)
Mark	94	0.396	0.208	0.475	0.340	0.575	0.500	0.855	0.717 (10)	0.871	0.704 (10)
McCallum	94	0.785	0.504	0.830	0.525	0.900	0.717	0.924	0.785 (5)	0.955	0.824 (10)
Mitchell	92	0.750	0.487	0.762	0.485	0.785	0.490	0.850	0.776 (20)	0.824	0.643 (20)
Mulford	94	0.555	0.322	0.573	0.305	0.853	0.727	0.911	0.826 (10)	0.926	0.833 (10)
Ng	87	0.750	0.542	0.785	0.575	0.915	0.845	0.951	0.925 (50)	0.953	0.911 (20)
Pereira	88	0.565	0.333	0.548	0.320	0.788	0.720	0.926	0.851 (5)	0.946	0.923 (5)
Voss	89	0.375	0.220	0.345	0.196	0.625	0.600	0.876	0.633 (10)	0.850	0.667 (10)
Mean	**90**	**0.596**	**0.340**	**0.611**	**0.371**	**0.792**	**0.647**	**0.909**	**0.756**	**0.911**	**0.765**

Table 3: Clustering results of the Web Appearances data set in terms of pair-level pairwise F1 Score(%) (F1P) and cluster-level pairwise F1 score(%) (F1C). Greedy Agglomerative Clustering is compared as a baseline approach. Our approaches (PLSA and LDA) consistently show better results than both spectral clustering and DBSCAN methods. The number of topics K is chosen from the set $\{2, 5, 10, 20, 50, 100, 200\}$. The best results with optimal K (given in parentheses) are presented here.

outperform both greedy agglomerative clustering and spectral clustering, and better than DBSCAN except for the *M. Jones* class. Overall, PLSA and LDA achieve 92.3% and 93.6% pair-wise F1 metric, respectively, which shows a gain of more than 40% and 86.6% improvement over the spectral clustering and greedy agglomerative clustering. DBSCAN also achieves a comparative result (89.3%) in this case.

In terms of the cluster F1 metric, PLSA and LDA models have almost the same performance, both achieve significantly better results (more than 140%) than spectral clustering and agglomerative clustering. The relatively high F1C scores of our methods indicate that the number of unique authors can be estimated with the number of achieved clusters from the original data set.

Illustrative examples of these results are presented in Table 5, which summarizes the results of the PLSA model by showing the 10 highest probability words along with their corresponding conditional probabilities from 4 topics in the CiteSeer data set. Additionally, we show 3 author name variations corresponding to the same person with their probability for each topic. The appearance of new authors is handled by using the "folding-in" process discussed in Section 3.3. Clearly, the selected 4 topics reveal that the 3 name variations have very high probability to be the same author. The figure beneath depicts the probability distributions over 50 topics, of which the three names exhibit quite similar patterns.

Likewise, Table 6 lists the results from the LDA model. We depict several topics that show the maximum differences in probabilities to disambiguate authors with *exactly* the same name. As for the name "Yang Song", one author has very high probability of topic 4 (0.2210) while the other are highly related with topic 11 (0.2682), thus showing completely different patterns of their probability distributions over topics. The same situation can also be observed for the author "Jun Yang" (see Table 7) by topic 40 ("Database") and topic 42 ("Multimedia Retrieval").

Topic 13 "Image Categorization"		**Topic 24 "Content Retrieval"**	
classifiers	0.0311	feature	0.0318
region	0.0285	learning	0.0216
image	0.0211	content	0.0138
indexing	0.0157	images	0.0130
photo	0.0152	clusters	0.0130
colors	0.0133	cluster	0.0130
color	0.0123	retrieval	0.0112
extract	0.0111	location	0.0112
aesthetics	0.0103	query	0.0064
light	0.0085	classifiers	0.0061
James Wang	0.2721	James Wang	0.1478
J. Z. Wang	0.2215	J. Z. Wang	0.1362
James Ze Wang	0.2533	James Ze Wang	0.1577

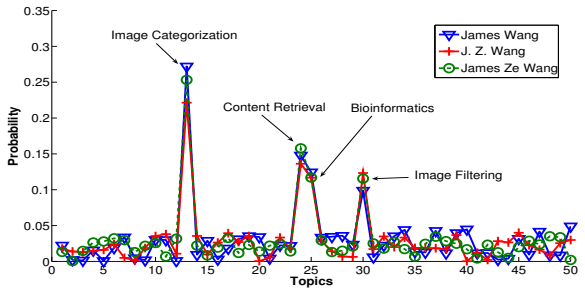
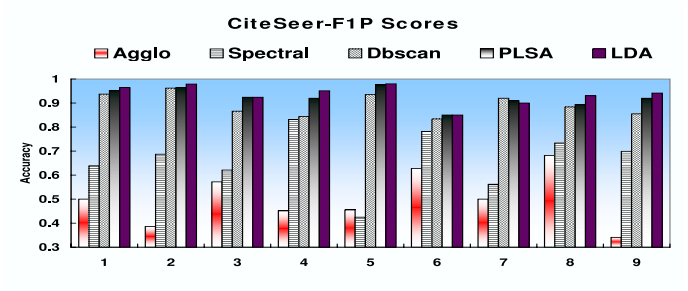


Table 5: An illustrative example of the author-topic relationships in the CiteSeer data set extracted by the topic-based PLSA model. 10 most corresponding words are shown for each topic. We summarize the titles of the topics to the best of our understanding. Below each topic shows the probabilities of authors with name variations. In this example three names refer to the same person.



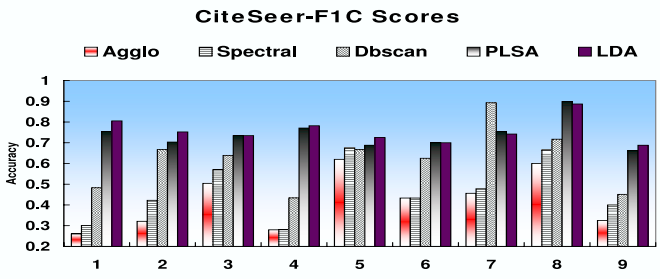


Figure 4: Clustering results on the CiteSeer data set. 1:A. Gupta, 2:A. Kumar, 3:C. Chen, 4:D. Johnson, 5:J. Robinson, 6:J. Smith, 7:K. Tanaka, 8:M. Jones, 9:M. Miller.

Topic 4 "Text Classification"		Topic 11 "Vision & Motion"	
boosting	0.0473	position	0.0486
text	0.0473	motion	0.0411
classification	0.0473	perceive	0.0220
classifiers	0.0473	vision	0.0220
feature	0.0422	label	0.0162
document	0.0215	tracked	0.0162
corpora	0.0215	moving	0.0111
words	0.0116	actions	0.0111
vectors	0.0116	humans	0.0105
dimensionality	0.0116	visual	0.0105
Yang Song(PSU)	0.2210	Yang Song(PSU)	0.0320
Yang Song(Caltech)	0.0202	Yang Song(Caltech)	0.2682

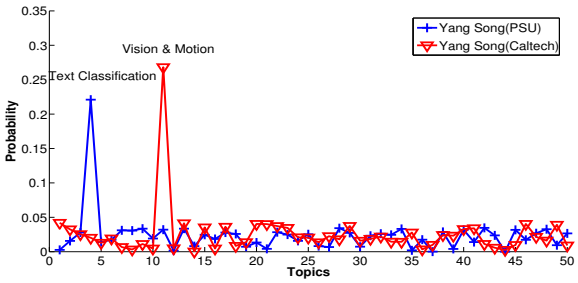


Table 6: LDA topic distributions of two authors with the same name "Yang Song".

Topic 40 "Database"		Topic 42 "Multimedia"	
query	0.0375	retrieval	0.0411
xml	0.0321	multimedia	0.0411
database	0.0321	broadcast	0.0360
scalability	0.0315	video	0.0311
process	0.0315	shot	0.0311
storage	0.0215	labeling	0.0311
memory	0.0215	flash	0.0215
performance	0.0113	learning	0.0215
structure	0.0113	visual	0.0215
generating	0.0113	show	0.0215
Jun Yang(Duke)	0.1258	Jun Yang(Duke)	0.0398
Jun Yang(CMU)	0.0477	Jun Yang(CMU)	0.2781

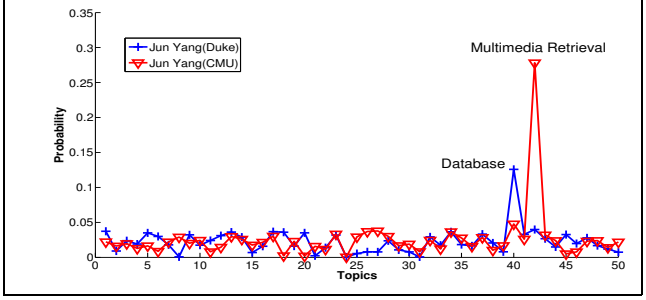


Table 7: LDA topic distributions of two authors with the same name "Jun Yang".

6.3.1 *Scalability and comparison of the two models*

Theoretical issue of scalability for large-scale data set has not yet been addressed for either PLSA or LDA. As a result, we empirically tested our models for the entire CiteSeer data set with more than 750,000 documents. PLSA yields 418,500 unique authors in 2,570 minutes, while LDA finishes in 4,390 minutes with 418,775 authors. Both are quite consistent with previous results [12, 9]. Considering that our methods only make use of a small portion of the text for each instance (metadata plus the first page), we believe the framework can be efficient for large-scale data sets.

The results of the two models are quite close to each other in both metrics across two data sets; however, they may have different generalization capabilities. In Figure 5, we show the comparison between PLSA and LDA in terms of the exponential of the negative likelihood (a.k.a. *perplexity*), which is commonly used as a measure of the generalization performance of probabilistic models. Generally, lower perplexity over a set of held-out test data indicates better performance.

Figure 5 depicts the results for the 2 models being compared. Both models exhibit the overfitting problem when the number of topics K increases. Comparatively, LDA is less sensitive to the change of K. This probably explains why PLSA is not a *fully generative* model, since PLSA applies "folding-in" process to manage new documents. This process assumes that documents in the testing set exhibit the same topic distribution (E-step of the EM algorithm) as those in the training set, which is not essentially true in many cases. In LDA, by generating probability with predefined priors to testing documents, all documents essentially exhibit the same topic distribution, thus no assumption is required for new authors in the testing documents.

Nevertheless, the best performance for both models are quite close, achieved when K is either 5 or 10.

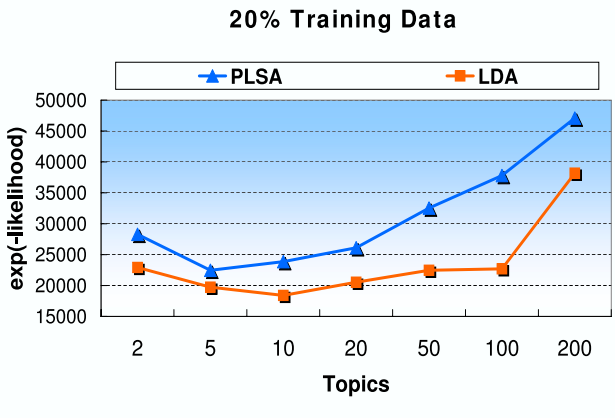


Figure 5: Exponential of the Negative Likelihood of the two models for the CiteSeer data set. X axis shows the number of topics. Here we show the results of using 20% training data.

7. CONCLUSION

We have proposed a novel framework for unsupervised name disambiguation by leveraging graphical Bayesian models and a hierarchical clustering method. Our approach has been demonstrated to be more effective than other unsupervised learning methods including spectral clustering and DBSCAN. A series of experiments were performed that verified the advantages of our approach on both web data and scientific documents. Although our primary focus in this paper is on person name disambiguation, our general approach should be equally applicable to other entity disambiguation domains. Potential applications include noun phrases disambiguation, e.g., "tiger" as an animal, "tiger" as a golf player, "tiger" the baseball team, "tiger" the operating system or "tiger" for the new Java version. And of course, it would be interesting to see whether our framework can be applied to automatic image annotation and other fields.

8. REFERENCES

[1] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 463–470, New York, NY, USA, 2005.

[2] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *SDM*, 2006.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[4] C. H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 625–632, New York, NY, USA, 2006. ACM Press.

[5] C. Charalambous. Maximum likelihood parameter estimation from incomplete data via the sensitivity equations: The continuous-time case, 1998.

[6] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[7] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: an automatic citation indexing system. In *DL '98: Proceedings of the third ACM conference on Digital libraries*, pages 89–98, New York, NY, USA, 1998. ACM Press.

[8] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *JCDL '04: Proceedings of the 4th ACM/IEEE joint conference on Digital libraries*, pages 296–305, New York, 2004.

[9] H. Han, H. Zha, and C. L. Giles. Name disambiguation in author citations using a k-way spectral clustering method. In *JCDL '05: Proceedings of the 5th ACM/IEEE joint conference on Digital libraries*, pages 334–343, New York, NY, USA, 2005. ACM Press.

[10] T. Hofmann. Probabilistic Latent Semantic Indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, Berkeley, California.

[11] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 259–266, New York, NY, USA, 2003.

[12] J. Huang, S. Ertekin, and C. L. Giles. Efficient name disambiguation for large-scale databases. In *the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 536–544. Springer-Verlag Berlin Heidelberg, 2006.

[13] W. J.H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[14] X. Jin, Y. Zhou, and B. Mobasher. Web usage mining based on probabilistic latent semantic analysis. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 197–205, New York, NY, USA, 2004. ACM Press.

[15] B. King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 69:86–101, 1967.

[16] D. Lee, B.-W. On, J. Kang, and S. Park. Effective and scalable solutions for mixed and split citation problems in digital libraries. In *IQIS '05: Proceedings of the 2nd international workshop on Information quality in information systems*, pages 69–76, New York, 2005.

[17] V. I. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Probl. Inform. Transmiss.*, 1:8–17, 1965.

[18] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 33–40, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[19] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[20] P. H. Sneath and R. R. Sokal. Numerical taxonomy. *Freeman, London, UK*, 1973.

[21] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, New York, NY, USA, 2006.

[22] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA, 2006. ACM Press.

[23] B. Wellner, A. McCallum, F. Peng, and M. Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 593–601, Arlington, Virginia, United States, 2004. AUAI Press.

[24] G. Xu, Y. Zhang, J. Ma, and X. Zhou. Discovering user access pattern based on probabilistic latent factor model. In *ADC '05: Proceedings of the sixteenth Australasian database conference*, pages 27–35, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.