A Latent Topic Model for Complete Entity Resolution

Liangcai Shu, Bo Long, Weiyi Meng

Department of Computer Science, SUNY at Binghamton
Binghamton, New York 13902, U. S. A.
lshu@cs.binghamton.edu
blong1@binghamton.edu
meng@cs.binghamton.edu

***Abstract*— In bibliographies like DBLP and Citeseer, there are three kinds of entity-name problems that need to be solved. First, multiple entities share one name, which is called the *name sharing* problem. Second, one entity has different names, which is called the *name variant* problem. Third, multiple entities share multiple names, which is called the *name mixing* problem. We aim to solve these problems based on one model in this paper. We call this task *complete entity resolution*. Different from previous work, our work use *global* information based on data with two types of information, *words* and *author names*. We propose a generative latent topic model that involves both author names and words — the LDA-dual model, by extending the LDA (Latent Dirichlet Allocation) model. We also propose a method to obtain model parameters that is global information. Based on obtained model parameters, we propose two algorithms to solve the three problems mentioned above. Experimental results demonstrate the effectiveness and great potential of the proposed model and algorithms.**

I. Introduction

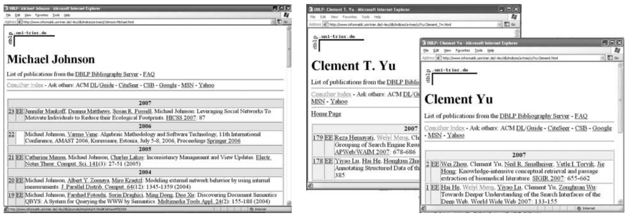
Bibliography websites like DBLP [1] and Citeseer provide much convenience for scientists and researchers to search or browse the citations of papers. A citation includes authors' names, paper title, venue of publication, publication year, etc. The information are very useful. But there are problems about the correctness of the information. Entities, particularly authors, are frequently inconsistent with their names. We identify three major entity-name problems — the *name sharing* problem, the *name variant* problem and the *name mixing* problem. These problems cause confusion for users of bibliography websites. Now we use real examples to explain these problems.

The name sharing problem

A full name may be shared by multiple authors. DBLP bibliography treats these authors as one entity and lists all citations under the shared name in a single page. And users of DBLP cannot distinguish those authors with the same full name. This causes confusion and handicaps the usage of this bibliography. We call this problem the *name sharing* problem or *multiple-entity-one-name* problem.

In Figure 1 (left), all 23 citations with the full name *Michael Johnson* are listed in a single web page at the DBLP website. But based on our investigation, citations listed under this name are from five different entities.

This problem has attracted much attention from researchers in recent years. Different names for this problem are name disambiguation [2][3], object distinction [4], mixed citation [5] and author disambiguation [6].



DBLP: Michael Johnson - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/j/Johnson:Michael.html
dblp.uni-trier.de

Michael Johnson

List of publications from the DBLP Bibliography Server - FAQ

Coauthor Index - Ask others: ACM DL/Guide - CiteSeer - CSB - Google - MSN - Yahoo

2007

23 EE Jennifer Mankoff, Deanna Matthews, Susan R. Fussell, Michael Johnson: Leveraging Social Networks To Motivate Individuals to Reduce their Ecological Footprints. HICSS 2007: 87

2006

22 Michael Johnson, Varmo Vene: Algebraic Methodology and Software Technology, 11th International Conference, AMAST 2006, Kuressaare, Estonia, July 5-8, 2006, Proceedings Springer 2006

2005

21 EE Catherine Menon, Michael Johnson, Charles Lakos: Inconsistency Management and View Updates. Electr. Notes Theor. Comput. Sci. 141(3): 27-51 (2005)

2004

20 EE Michael Johnson, Albert Y. Zomaya, Miro Kraetzl: Modeling external network behavior by using internal measurements. J. Parallel Distrib. Comput. 64(12): 1345-1359 (2004)

19 EE Michael Johnson, Farshad Fotouhi, Sorin Draghici, Ming Dong, Duo Xu: Discovering Document Semantics QBYS: A System for Querying the WWW by Semantics. Multimedia Tools Appl. 24(2): 155-188 (2004)

http://www.informatik.uni-trier.de/~ley/db/journals/mta/mta24.html#JohnsonFDDX04
Internet
DBLP: Clement T. Yu - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/y/Yu:Clement_T=.html
dblp.uni-trier.de

Clement T. Yu

List of publications from the DBLP Bibli

Coauthor Index - Ask others: ACM DL/Gu
MSN - Yahoo

Home Page

2007

179 EE Reza Hemayati, Weiyi Meng, Cle
Grouping of Search Engine Resu
APWeb/WAIM 2007: 678-686

178 EE Yiyao Lu, Hai He, Hongkun Zha
Annotating Structured Data of th
385

DBLP: Clement Yu - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/y/Yu:Clement.html
dblp.uni-trier.de

Clement Yu

List of publications from the DBLP Bibliography Server - FAQ

Coauthor Index - Ask others: ACM DL/Guide - CiteSeer - CSB - Google - MSN - Yahoo

2007

2 EE Wei Zhou, Clement Yu, Neil R. Smalheiser, Vetle I. Torvik, Jie Hong: Knowledge-intensive conceptual retrieval and passage extraction of biomedical literature. SIGIR 2007: 655-662

1 EE Hai He, Weiyi Meng, Yiyao Lu, Clement Yu, Zonghuan Wu: Towards Deeper Understanding of the Search Interfaces of the Deep Web. World Wide Web 2007: 133-155

Fig. 1. An example of the *name sharing* problem (left) and an example of *name variant* problem (right) in DBLP.

The name variant problem

The second problem is called the *name variant* problem or *one-entity-multiple-name* problem. An author may have multiple full names which are called *name variants* of this author. But DBLP treats these full names as from different entities and lists them in separate web pages. For example, in Figure 1 (right), *Clement Yu* and *Clement T. Yu* represent the same entity. But DBLP treats them as the names of two entities and lists the citations in separate web pages.

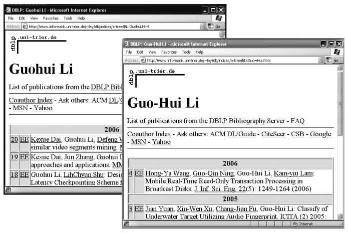
This problem has been also referred to as the entity resolution [7] or split citation [5] problem. And it has been researched for quite a few years because of its strong connection with data cleaning and record linkage.

The name mixing problem

The two problems above may intertwine. In that case, multiple entities share multiple names. This is called the *name mixing* problem or *multiple-name-multiple-entity* problem. For example, in Figure 2, four entities share the name *Guohui Li* and two of them share another name *Guo-Hui Li*. But citations under *Guohui Li* and those under *Guo-Hui Li* appear in separate web pages in DBLP. Solving this problem requires solving name sharing problem and name variant problem simultaneously.

Challenges and potential solutions

The three problems are very challenging. First, authors may move from one place to another, so their coauthors may change significantly. That makes it difficult to integrate citations before and after their moves. Second, authors may change research interests and collaborate with a wide variety of other researchers. If they share identical names, it is difficult to differentiate their citations. Third, the two scenarios above


DBLP: Guohui Li - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/l/Li:Guohui.html
Go
dblp.uni-trier.de
DBLP: Guo-Hui Li - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/l/Li:Guo=Hui.html
Go
dblp.uni-trier.de
Internet

Guohui Li

List of publications from the DBLP Bibl

Coauthor Index - Ask others: ACM DL/
- MSN - Yahoo

2006		
20	EE	Kexue Dai, Guohui Li, Defeng W similar video segments mining. M
19	EE	Kexue Dai, Jun Zhang, Guohui L approaches and applications. MM
18	EE	Guohui Li, LihChyun Shu: Desig Latency Checkpointing Scheme f

Guo-Hui Li

List of publications from the DBLP Bibliography Server - FAQ

Coauthor Index - Ask others: ACM DL/Guide - CiteSeer - CSB - Google - MSN - Yahoo

2006		
4	EE	Hong-Ya Wang, Guo-Qin Ning, Guo-Hui Li, Kam-yiu Lam: Mobile Real-Time Read-Only Transaction Processing in Broadcast Disks. J. Inf. Sci. Eng. 22(5): 1249-1264 (2006)
2005		
3	EE	Jian Yuan, Xin-Wen Xu, Chang-Jian Fu, Guo-Hui Li: Classify of Underwater Target Utilizing Audio Fingerprint. ICITA (2) 2005:

Fig. 2. An example of the *name mixing* problem.

may happen simultaneously and cause more intricacies. Furthermore, for the *name mixing* problem with the two scenarios, intricacies would be very extensive.

We try to see through these intricacies and describe citations and documents by something invariant, the *global* information for a corpus or a bibliography. We explain it with the help of an example of two citations.

- Clement T. Yu. Multimedia Metasearch Engines. *Multimedia Information Systems* 2002: 5-6
- Fang Liu, Shuang Liu, Clement T. Yu, Weiyi Meng, Ophir Frieder, and David Grossman. Database selection in intranet mediators for natural language queries. *CIKM* 2005: 229-230

In this example, to solve the name sharing problem, we try to find out whether the name *Clement T. Yu* in the two citations are from the same person. The two citations above have just one author name *Clement T. Yu* in common. They seems to be very different. But actually the two citations have close relationship. First, *database selection* and *queries* are both important concepts of *metasearch engines*. Second, *Clement T. Yu* and *Weiyi Meng* have appeared in many citations about *metasearch engines*.

Inspired by the LDA model [8], we try to catch global information by *topic*. we propose the LDA-dual model in which we extend the notion *topic* in LDA. While it is only a Dirichlet distribution over words in the LDA, a topic is represented by two Dirichlet distributions in the LDA-dual, one over words and the other over author names. Then a topic is characterized by a series of words and a series of author names. For example, topic *physics* might be characterized by "relativity" and "Einstein", and other words and names.

Topics come from the whole corpus, so we call them *global* information. Author names, the paper title, etc for a citation are called *local* information. The two citations above have little local information in common, but they probably have much *global* information in common that helps estimate similarity of two citations.

Complete entity resolution

We propose the notion of *complete entity resolution* to refer to the three entity-name problems in a bibliography. Previous work does not solve the three problems simultaneously.

Work in [7] is aimed at entity resolution. It is aimed at solving only the name variant problem and using only author names in citations. In this paper, we solve name sharing and name variant problem, and use both author names and words, both global and local information of documents.

We first learn the LDA-dual model to grasp global information, for representing a citation by a vector that is a mixture of topics. The topic similarity of two citations is computed based on two vectors. Other similarities are computed based on local information like coauthor names, titles, etc. Based on these similarities, a classifier is used to decide if two citations are from the same person. And based on the classifier, we propose two algorithms to solve the name sharing problem and the name variant problem.

Contributions

Our contributions in this paper are summarized below.

• We propose a generative model, LDA-dual. Different from previous models, the model involves two types of data — words and author names.

• We propose an approach to obtain the model parameters for a corpus. And we deduce a series of conditional probabilities in support of *Gibbs sampling*. Different from previous work, hyperparameter vectors α, β and γ are also included in the sampling process by means of *Metropolis-Hasting within Gibbs*.

• Based on the model parameters, we propose two algorithm to solve the name sharing problem and the name variant problem.

The rest of the paper is organized as follows. In Section II we propose the LDA-dual model. Section III presents how to learn model parameters by Gibbs sampling. Section IV discusses LDA-dual's application in entity, including two algorithms for the name sharing and name variant problems. Section V reports experimental results. Section VI is related work. Section VII is conclusion. Appendix is proofs of propositions.

II. The LDA-dual model

To obtain global information of a corpus, we propose a generative latent topic model that we called the LDA-dual model as shown in Figure 3. The explanations to the symbols are included in Table I. This is a kind of *graphical model* [9]. There is a conditional dependence assumption for this kind of model, i.e., each node in Figure 3 is conditionally independent of its non-descendants, given its immediate parents [10]. The assumption is the basis for most derived conditional probabilities in Sections II and III in this paper.

The LDA-dual model is extended from the Latent Dirichlet Allocation (LDA) model [8]. The LDA model has two basic assumptions. First, every document is a mixture of topics. Second, every topic is a Dirichlet distribution over words in the vocabulary. The LDA model has been widely applied to analyzing documents that include one type of information like words. But It has not been applied to documents that contain two or more types of information, such as a bibliography that includes both words and author names, a demographic document that includes words, person names and place names, etc. To apply to documents containing author names as well as words, we extend LDA in our model by adding the third assumption — every topic is a Dirichlet distribution over all author names.

In our model, words and author names are generated in a parallel manner for a document. Specifically, given the number of words and the number of author names, this model generates every word by first choosing a topic and then choosing a

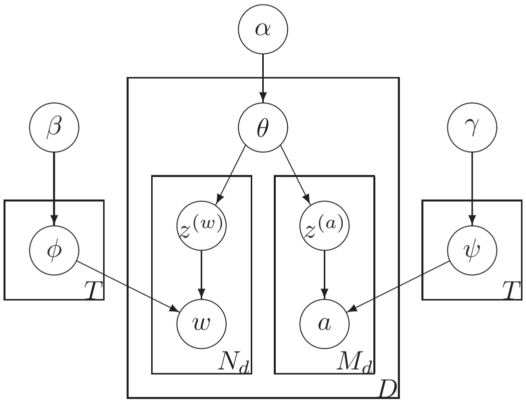


Fig. 3. The LDA-dual model. w and a are observed variables. The notations of the model are given in Table I.

TABLE I
NOTATIONS

Symbols	Meanings
$\mathcal{W}, W$	the vocabulary $\mathcal{W}$ with size W
$\mathcal{A}, A$	the author name list $\mathcal{A}$ with size A
$\mathcal{T}, T$	the set of topics $\mathcal{T}$ with size T
$\mathcal{D}, D$	a corpus $\mathcal{D}$ including D documents
N, M	N: # of *words*, M: # of *author names*, in $\mathcal{D}$
N_d	# of *words* in the dth document in $\mathcal{D}$
M_d	# of *author names* in the dth document in $\mathcal{D}$
α, β, γ	Dirichlet prior vectors, T-dimensional, W-dimensional and A-dimensional, respectively
θ	T-dimensional vector, a mixture of topics for a document
ϕ	W-dimensional vector, a topic's probability distribution over all words in $\mathcal{W}$
ψ	A-dimensional vector, a topic's probability distribution over all author names in $\mathcal{A}$
Θ	$D \times T$ matrix for $\mathcal{D}$, each row vector is a θ
Φ	$T \times W$ matrix, each row vector is a ϕ
Ψ	$T \times A$ matrix, each row vector is a ψ
w, a	w is a word, and a is an author name
t	a topic in $\mathcal{T}$
$\mathbf{d}$	a document. $\mathbf{d}^{(w)}$ is its word's part and $\mathbf{d}^{(a)}$ is its author name's part
$z^{(w)}$	a sample of topic for a word
$z^{(a)}$	a sample of topic for an author name
$\mathbf{z}$	a topic assignment for a document. $\mathbf{z}^{(w)}$ is its word's part and $\mathbf{z}^{(a)}$ is its author name's part
$\mathcal{Z}$	a topic assignment for a corpus. $\mathcal{Z}^{(w)}$ is its word's part and $\mathcal{Z}^{(a)}$ is its author name's part
n_{dt}, n_{tw}	counts in $\mathcal{Z}$ for *words*
m_{dt}, m_{ta}	counts in $\mathcal{Z}$ for *author names*

word according to the topic's Dirichlet distribution over words. And it generates every author name by first choosing a topic and then choosing an author name according to the topic's Dirichlet distribution over author names. The latent variable *topic* is slightly different from that in LDA. To understand that easily, we can think of a topic as a research group, a group of people who share research interests and coauthor papers, because a research group includes information for both words and author names.

In Section III, we learn this model by Gibbs sampling. In Section IV, we present an application of this model to complete entity resolution. After all, this model can also be applied to other domains in which documents include two types of information.

A. Documents

We first define documents that include two types of information. It is called *words* and *author names* in this paper, but it can have other names for different domains.

Definition 1: A *document* is a collection of N words and M author names, denoted as $\mathbf{d} = \{w_1, \ldots, w_N, a_1, \ldots, a_M\}$, where w_n is the *n*th word and a_m the *m*th author name. And we denote the collection of words as $\mathbf{d}^{(w)} = \{w_1, \ldots, w_N\}$ and the collection of author names as $\mathbf{d}^{(a)} = \{a_1, \ldots, a_M\}$.

A *corpus* is a collection of D documents, denoted by $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_D\}$. The words part of $\mathcal{D}$ is $\mathcal{D}^{(w)} = \{\mathbf{d}_1^{(w)}, \ldots, \mathbf{d}_D^{(w)}\}$ and the author names part is $\mathcal{D}^{(a)} = \{\mathbf{d}_1^{(a)}, \ldots, \mathbf{d}_D^{(a)}\}$.

Definition 2: A *sample of topic* is a topic that is drawn from the set of topics $\mathcal{T}$ and corresponds to a word or an author name in a document. It is denoted by $z^{(w)}$ if corresponding to a word, or $z^{(a)}$ if corresponding to an author name. A *sample of topic* can be considered as a mapping from a word or an author name to a topic.

B. Generative process

Two probability distributions are involved in this model — the Dirichlet distribution and the multinomial distribution.

(a). Dirichlet distribution

In the LDA model [8] and the LDA-dual model, we obtain a mixture of topics, denoted by θ, by drawing a sample from a Dirichlet distribution $p(\theta|\alpha) = \frac{\Gamma(\sum_{t=1}^{T}\alpha_t)}{\prod_{t=1}^{T}\Gamma(\alpha_t)}\prod_{t=1}^{T}\theta_t^{\alpha_t-1}$. Here θ is a T-dimensional vector and $\sum_{t=1}^{T}\theta_t = 1$, where θ_t is the tth component of θ. In this paper, hyperparameter α is a vector and called Dirichlet prior vector. Here $\alpha_t > 0$, where α_t is the tth component of α. $\Gamma(\cdot)$ is the Gamma function [11].

Similarly, ϕ is drawn from a Dirichlet distribution with hyperparameter β. ψ is drawn from a Dirichlet distribution with hyperparameter γ.

(b). Multinomial distribution

In the LDA model [8][12], given a mixture of topics θ, *one* topic is drawn from a probability distribution. This distribution is a special case of the multinomial distribution of n independent trials, with $n = 1$. A word is drawn from the multinomial distribution with the parameter ϕ. In addition, in the LDA-dual model, an author name is drawn from the multinomial distribution with the parameter ψ.

The Dirichlet distribution is conjugate prior of the multinomial distribution. This relationship helps integrate some intermediate latent variables like θ, ϕ and ψ in parameter estimation in Section III.

Based on the Dirichlet distribution and multinomial distribution, this model involves several generative steps which are represented by nodes and arrows between nodes as shown in Figure 3. Steps 1 and 4 below appeared in [8], and Steps 2 and 6 appeared in [12]. Steps 3, 5 and 7 are introduced in this model.

1. $\theta|\alpha \sim Dirichlet(\alpha)$:

For a document, choose a mixture of topics θ from a Dirichlet distribution with hyperparameter vector α. For a corpus of D documents, there are D such θ's and they construct a $D \times T$ matrix, denoted by Θ. Each row of Θ is a θ corresponding to a document. θ_d denotes the dth row that corresponds to the dth document.

2. $\phi|\beta \sim Dirichlet(\beta)$:

For a topic, choose ϕ from a Dirichlet distribution over words with hyperparameter vector β. Since we have T topics altogether, there are T such ϕ's and they construct a $T \times W$ matrix, denoted by Φ. Each row of Φ is a ϕ corresponding to a topic. ϕ_t denotes the tth row that corresponds to the tth topic.

3. $\psi|\gamma \sim Dirichlet(\gamma)$:

For a topic, choose ψ from a Dirichlet distribution over author names with hyperparameter vector γ. There are T such ψ's and they construct a $T \times A$ matrix, denoted by Ψ. Each row of Ψ is a ψ corresponding to a topic. ψ_t denotes the tth row that corresponds to the tth topic.

4. $z^{(w)}|\theta \sim Multinomial(\theta)$:

For a word in a document, choose a sample of topic $z^{(w)}$ given θ, a mixture of topic for the document. This is a multinomial process.

5. $z^{(a)}|\theta \sim Multinomial(\theta)$:

For an author name in a document, choose a sample of topic $z^{(a)}$ given θ, a mixture of topic for the document.

6. $w|z^{(w)}, \phi \sim Multinomial(\phi)$:

Choose a word w, given a sample of topic $z^{(w)}$ and its corresponding ϕ, the distribution over words for topic $z^{(w)}$.

7. $a|z^{(a)}, \psi \sim Multinomial(\psi)$:

Choose an author name a, given a sample of topic $z^{(a)}$ and its corresponding ψ, the distribution over author names for topic $z^{(a)}$.

Definition 3: A *topic assignment* $\mathbf{z}$ of a document $\mathbf{d}$ is the set of samples of topics for this document. We define $\mathbf{z}^{(w)} = \{z_1^{(w)}, \ldots, z_N^{(w)}\}$ and $\mathbf{z}^{(a)} = \{z_1^{(a)}, \ldots, z_M^{(a)}\}$. Then we have $\mathbf{z}$ = $\mathbf{z}^{(w)} \cup \mathbf{z}^{(a)}$. $\mathbf{z}$ is called the *total topic assignment*, and $\mathbf{z}^{(w)}$ and $\mathbf{z}^{(a)}$ are called *partial topic assignments*.

Definition 4: A *topic assignment* $\mathcal{Z}$ is a set of samples of topic for a corpus $\mathcal{D}$. We define $\mathcal{Z} = \bigcup_{d=1}^{D} \mathbf{z}_d$, $\mathcal{Z}^{(w)} = \bigcup_{d=1}^{D} \mathbf{z}_d^{(w)}$ and $\mathcal{Z}^{(a)} = \bigcup_{d=1}^{D} \mathbf{z}_d^{(a)}$, where $\mathbf{z}_d$, $\mathbf{z}_d^{(w)}$ and $\mathbf{z}_d^{(a)}$ are topic assignments of the dth document in $\mathcal{D}$. $\mathcal{Z}$ is called the *total topic assignment* of $\mathcal{D}$. $\mathcal{Z}^{(w)}$ and $\mathcal{Z}^{(a)}$ are called the *partial topic assignment* of $\mathcal{D}$.

A *topic assignment* can be understood as a function that maps from a collection of words and author names to the set of topics, while the *sample of topic* is a mapping from a word or an author name to a topic.

C. An illustrative example

Here is an example for explanation. This is a case of the *name mixing* problem. Two entities share two names — *George Bush* and *George W. Bush*.

Example 1: A tiny corpus includes only two documents. Document #1 includes three citations under *George Bush*:

1.1 George Bush, and D. Cheney: *The Iraq war*.
1.2 George Bush, and C. Powell: *The Gulf war*.
1.3 George Bush: *The Gulf report*.

Document #2 includes one citation under *George W. Bush*:

2.1 George W. Bush, and D. Cheney: *The Iraq report*.

The word "*The*" is treated as a stop word and eliminated. The number of words in the corpus $N = 8$. The number of author names in the corpus $M = 7$. The counts are based on the number of occurrences.

The vocabulary $\mathcal{W}$ = {*Gulf, Iraq, report, war*} with size W = 4. The author name list $\mathcal{A}$ = {*George Bush, George W. Bush, D. Cheney, C. Powell*} with size A = 4. Let the set of topics $\mathcal{T}$ = {1, 2}, where we use its order number to represent a topic. T = 2 is the size of $\mathcal{T}$.

D. The probability of a corpus

We intend to derive the conditional probability of corpus $\mathcal{D}$ given α, β and γ to see if we can estimate parameters by EM algorithms. We can obtain the probability of a document $\mathbf{d}$, by summing over $\mathbf{z}$ and integrating over θ, Φ, Ψ:

$$p(\mathbf{d}|\alpha, \beta, \gamma) = \int_\theta p(\theta|\alpha)$$
$$\left\{ \prod_{i=1}^{N_d} \sum_{z_i^{(w)} \in \mathcal{T}} p(z_i^{(w)}|\theta) \int p(w_i|z_i^{(w)}, \phi_{z_i^{(w)}}) p(\phi_{z_i^{(w)}}|\beta) \, d\phi_{z_i^{(w)}} \right\}$$
$$\left\{ \prod_{j=1}^{M_d} \sum_{z_j^{(a)} \in \mathcal{T}} p(z_j^{(a)}|\theta) \int p(a_j|z_j^{(a)}, \psi_{z_j^{(a)}}) p(\psi_{z_j^{(a)}}|\gamma) \, d\psi_{z_j^{(a)}} \right\} d\theta. \tag{1}$$

Based on the probability of a document, we obtain the probability of a corpus:

$$p(\mathcal{D}|\alpha, \beta, \gamma) = \prod_{d=1}^{D} p(\mathbf{d}_d|\alpha, \beta, \gamma). \tag{2}$$

The integrals in Formula (1) are intractable [13] [8], due to the coupling of $\mathbf{z}^{(w)}$ and ϕ, and that of $\mathbf{z}^{(a)}$ and ψ. So It is quite difficult to estimate parameters α, β and γ by EM algorithms.

Fortunately, some approximate inference algorithms could be used in this case, such as variational approximation [8][14], expectation propagation [15] and Monte Carlo simulation [13][16][17][18][19][20].

III. Gibbs Sampling

In this paper, we use Gibbs sampling [13][18][21][20], a Monte Carlo simulation algorithm, to estimate parameters of the LDA-dual model. Gibbs sampling simulates a multivariate probabilistic distribution by only considering univariate conditional distributions. The key idea is to sample one random variable when other random variables are assigned fixed values. And each random variable is sampled in this way alternately. Such univariate conditional distributions are far easier and faster to simulate than overall distribution.

We extended the work in [7][12]. [7][12] used Gibbs Sampling for the LDA model. In their work, Dirichlet priors like α are single values, instead of vectors. Also, Dirichlet priors are fixed and excluded from sampling process. Furthermore, their sampling process involves only one kind of information — words [12] or author names [7].

Instead, in our sampling process, two types of data, both words and author names, are considered in our algorithm. And Dirichlet priors α, β and γ are vectors and they are included in the sampling process as well as samples of topic. The method of *Metropolis-Hasting within Gibbs* [17] is employed to sample Dirichlet prior vectors.

A. Posterior distribution

In order to estimate model parameters, we simulate the *posterior distribution* of model parameters given a corpus. We do not need to include all model parameters in the distribution, some parameters can be eliminated. Specifically, θ can be integrated out, because the Dirichlet distribution of θ is conjugate prior of the multinomial distribution of $\mathcal{Z}$. For the similar reason, ϕ and ψ can be integrated out.

Then we just need the *posterior distribution* of α, β, γ and topic assignment $\mathcal{Z}$, given a corpus $\mathcal{D}$. That is the critical part for Gibbs sampling.

According to generative processes of the LDA-dual model in Figure 3, considering the conditional independence assumption, we have the joint probability for a document: $p(\mathbf{z}, \mathbf{d}, \alpha, \beta, \gamma) = p(\alpha)\ p(\beta)\ p(\gamma)\ p(\mathbf{z}|\alpha)\ p(\mathbf{d}^{(w)}|\mathbf{z}^{(w)}, \beta)\ p(\mathbf{d}^{(a)}\ |\mathbf{z}^{(a)}, \gamma)$. In a similar way, we get the joint probability for a corpus: $p(\mathcal{Z}, \mathcal{D}, \alpha, \beta, \gamma) = p(\alpha)p(\beta)p(\gamma)p(\mathcal{Z}|\alpha)p(\mathcal{D}^{(w)}|\mathcal{Z}^{(w)}, \beta)p(\mathcal{D}^{(a)}|\mathcal{Z}^{(a)}, \gamma)$. And then we obtain the posterior distribution for Gibbs Sampling, the joint probability of $\mathcal{Z}$, α, β and γ given a corpus:

$$\begin{aligned} p(\mathcal{Z}, \alpha, \beta, \gamma|\mathcal{D}) = p(\alpha)\, p(\beta)\, p(\gamma) \\ p(\mathcal{Z}|\alpha)\, p(\mathcal{D}^{(w)}|\mathcal{Z}^{(w)}, \beta)\, p(\mathcal{D}^{(a)}|\mathcal{Z}^{(a)}, \gamma)\, /\, p(\mathcal{D}). \end{aligned} \tag{3}$$

Here $p(\mathcal{D})$ can be considered as a normalizing constant. In the following propositions, we determine $p(\mathcal{Z}|\alpha)$, $p(\mathcal{D}^{(w)}|\mathcal{Z}^{(w)}, \beta)$ and $p(\mathcal{D}^{(a)}|\mathcal{Z}^{(a)}, \gamma)$.

For convenience, we use an order number to refer to a topic, a document, a word in the vocabulary, or an author name in the name list thereafter. For example, topic t is the tth topic in the set of topics $\mathcal{T}$.

Proposition 1: Suppose $\mathcal{D}$ is a corpus of D documents, $\mathcal{Z}$ is a total topic assignment of $\mathcal{D}$, and α is a Dirichlet prior vector in the LDA-dual model. Then the probability of $\mathcal{Z}$ given α is

$$p(\mathcal{Z}|\alpha) = \prod_{d=1}^{D} \frac{\prod_{t=1}^{T} \prod_{k=1}^{n_{dt}+m_{dt}} (k-1+\alpha_t)}{\prod_{k=1}^{N_d+M_d} (k-1+\sum_{t=1}^{T} \alpha_t)}, \tag{4}$$

where α_t is the tth component of vector α, T is the number of topics, N_d is the number of words in document d, M_d is the number of author names in document d, n_{dt} is the number of times topic t is assigned to words in document d according to topic assignment $\mathcal{Z}$, and m_{dt} is the number of times topic t is assigned to author names in document d according to $\mathcal{Z}$.

Proposition 2: Suppose $\mathcal{D}$ is a corpus, $\mathcal{D}^{(w)}$ is the collection of words in $\mathcal{D}$, $\mathcal{Z}^{(w)}$ is a partial topic assignment of $\mathcal{D}$ for words, and β is a Dirichlet prior vector in the LDA-dual model. Then the probability of $\mathcal{D}^{(w)}$, the words part of $\mathcal{D}$, given $\mathcal{Z}^{(w)}$ and β, is

$$p(\mathcal{D}^{(w)}|\mathcal{Z}^{(w)}, \beta) = \prod_{t=1}^{T} \frac{\prod_{w=1}^{W} \prod_{k=1}^{n_{tw}} (k-1+\beta_w)}{\prod_{k=1}^{n_t} (k-1+\sum_{w=1}^{W} \beta_w)}, \tag{5}$$

where β_w is the wth component of vector β, T is the number of topics, W is the size of the vocabulary $\mathcal{W}$, n_{tw} is the number of times topic t is assigned to word w in the vocabulary according to the topic assignment $\mathcal{Z}^{(w)}$, and n_t is the number of times topic t is assigned to words according to $\mathcal{Z}^{(w)}$.

Proposition 3: Suppose $\mathcal{D}$ is a corpus, $\mathcal{D}^{(a)}$ is the collection of author names in $\mathcal{D}$, $\mathcal{Z}^{(a)}$ is a partial topic assignment of $\mathcal{D}$ for author names, and γ is a Dirichlet prior vector in the LDA-dual model. Then the probability of $\mathcal{D}^{(a)}$, the author names part of $\mathcal{D}$, given $\mathcal{Z}^{(a)}$ and γ, is

$$p(\mathcal{D}^{(a)}|\mathcal{Z}^{(a)}, \gamma) = \prod_{t=1}^{T} \frac{\prod_{a=1}^{A} \prod_{k=1}^{m_{ta}} (k-1+\gamma_a)}{\prod_{k=1}^{m_t} (k-1+\sum_{a=1}^{A} \gamma_a)}, \tag{6}$$

where γ_a is the ath component of vector γ, T is the number of topics, A is the size of the author name list $\mathcal{A}$, m_{ta} is the number of times topic t is assigned to author name a in the author name list, according to the topic assignment $\mathcal{Z}^{(a)}$, and m_t is the number of times topic t is assigned to author names according to $\mathcal{Z}^{(a)}$.

The proof of Proposition 1 is in Appendix. We do not include proofs of Propositions 2 and 3 due to space limit. In Formulas (4), (5) and (6), D, T, W, A, N_d and M_d are constants. n_{dt} and m_{dt} are counts in $\mathcal{Z}$. n_{tw} and n_t are counts in $\mathcal{Z}^{(w)}$. m_{ta} and m_t are counts in $\mathcal{Z}^{(a)}$.

After putting Formulas (4), (5) and (6) into Formula (3), Formula (3) can be used to derive univariate conditional distributions that can be directly used in Gibbs sampling.

B. Univariate conditional distributions

1) Conditional distribution of a sample of topic: Now we find out the conditional distribution of a sample of topic, for a word or an author name, over topics.

Considering the sample of topic $z_i^{(w)}$ for a word i in document d of corpus $\mathcal{D}$, we want to find $z_i^{(w)}$'s conditional distribution over topics. Let $\mathcal{Z}_{-i}^{(w)} = \mathcal{Z}^{(w)} \setminus \{z_i^{(w)}\}$, where $\setminus$ is set difference. We consider the conditional probability $p(z_i^{(w)} = t|\mathcal{Z}_{-i}^{(w)}, \mathcal{Z}^{(a)}, \alpha, \beta, \gamma, \mathcal{D})$ based on Formulas (3), (4) and (5). $\mathcal{Z}_{-i}^{(w)}$, $\mathcal{Z}^{(a)}$, α, β and γ are assumed to be fixed. Therefore in Formula (4), as the topic t changes, only n_{dt} changes. So for different t, the corresponding different factor in Formula (4) is $n_{dt} + m_{dt} - 1 + \alpha_t$, i.e., $n_{dt}^{(-i)} + m_{dt} + \alpha_t$, where $n_{dt}^{(-i)}$ is computed based on $\mathcal{Z}_{-i}^{(w)}$. Similarly, in Formula (5), as the topic t changes, only n_{tw} and n_t change. So for different t, the corresponding different factors in Formula (5) are $n_{tw} - 1 + \beta_w$ (i.e., $n_{tw}^{(-i)} + \beta_w$) and $n_t - 1 + \sum_{w=1}^{W} \beta_w$ (i.e., $n_t^{(-i)} + \sum_{w=1}^{W} \beta_w$).

In summary, based on Formulas (3), (4) and (5), we have the conditional distribution of the sample of topic $z_i^{(w)}$ for word i in document d:

$$p(z_i^{(w)} = t|\mathcal{Z}_{-i}^{(w)}, \mathcal{Z}^{(a)}, \alpha, \beta, \gamma, \mathcal{D}) \propto \frac{n_{tw}^{(-i)} + \beta_w}{n_t^{(-i)} + \sum_{k=1}^{W} \beta_k}(n_{dt}^{(-i)} + m_{dt} + \alpha_t) \tag{7}$$

Formula (7) agrees with our intuition. On its right side, the fraction means word i's weight in topic t, and $n_{dt}^{(-i)} + m_{dt} + \alpha_t$ is proportional to the topic t's weight in document d.

Similarly, based on Formulas (3), (4) and (6), for author name j in document d of corpus $\mathcal{D}$, to draw a topic t in the set of topics $\mathcal{T}$, we have the conditional distribution of the sample of topic $z_j^{(a)}$ for author name j:

$$p(z_j^{(a)} = t|\mathcal{Z}^{(w)}, \mathcal{Z}_{-j}^{(a)}, \alpha, \beta, \gamma, \mathcal{D}) \propto \frac{m_{ta}^{(-j)} + \gamma_a}{m_t^{(-j)} + \sum_{k=1}^{A} \gamma_k}(n_{dt} + m_{dt}^{(-j)} + \alpha_t), \tag{8}$$

where $\mathcal{Z}_{-j}^{(a)} = \mathcal{Z}^{(a)} \setminus \{z_j^{(a)}\}$, and $m_{ta}^{(-j)}$, $m_t^{(-j)}$ and $m_{dt}^{(-j)}$ are computed based on $\mathcal{Z}_{-j}^{(a)}$.

The sampling method for a sample of topic is straightforward because it is trivial to draw a topic from the conditional distributions in Formulas (7) and (8).

2) Conditional distribution of a component of Dirichlet prior vectors: We also include Dirichlet prior vectors α, β and γ into our sampling process. Each time we sample one component of a Dirichlet prior vector, assuming other components and parameters are fixed.

a) For a component of α: We consider the prior distribution of α_t, the tth component of α. [22] claims that log-normal can be used when "mean values are low, variances are large, and values cannot be negative". For this reason, the prior distribution $p(\alpha_t)$ is assumed to be a log-normal distribution over $(0, +\infty)$. Then $\ln \alpha_t$ is normally distributed and we assume the mean as 0 and the variance as σ^2. Then we have $p(\alpha_t = x) \propto \frac{1}{x}\exp(-\frac{\ln^2 x}{2\sigma^2})$. In Formula (4), we consider α_t, the tth component of α, assuming other parameters are fixed. As α_t changes, only $\prod_{k=1}^{n_{dt}+m_{dt}}(k - 1 + \alpha_t)$ and $\prod_{k=1}^{N_d+M_d}(k - 1 + \sum_{t=1}^{T} \alpha_t)$ are affected for each document.

Based on analysis above and Formulas (3) and (4), we have the conditional probability of α_t:

$$p(\alpha_t = x|\mathcal{Z}, \alpha^{(-t)}, \beta, \gamma, \mathcal{D}) \;\propto\; \frac{1}{x}\exp(-\frac{\ln^2 x}{2\sigma^2}) \prod_{d=1}^{D} \frac{\prod_{k=1}^{n_{dt}+m_{dt}}(k - 1 + x)}{\prod_{k=1}^{N_d+M_d}(k - 1 + x + \sum_{i=1}^{T-1} \alpha_i^{(-t)})} \tag{9}$$

where $x > 0$ is a real number, $\alpha^{(-t)}$ is the remaining vector after removing the tth component α_t from α, and $\alpha_i^{(-t)}$ is the ith component of vector $\alpha^{(-t)}$.

b) For a component of β: Similarly, for β_w, the wth component of β, assuming the prior distribution $p(\beta_w)$ is the same log-normal distribution as $p(\alpha_t)$, based on Formulas (3) and (5), we have the conditional probability of β_w, β's wth component:

$$p(\beta_w = x|\mathcal{Z}, \alpha, \beta^{(-w)}, \gamma, \mathcal{D}) \;\propto\; \frac{1}{x}\exp(-\frac{\ln^2 x}{2\sigma^2}) \prod_{t=1}^{T} \frac{\prod_{k=1}^{n_{tw}}(k - 1 + x)}{\prod_{k=1}^{n_t}(k - 1 + x + \sum_{i=1}^{W-1} \beta_i^{(-w)})} \tag{10}$$

From the formula above, we have interesting observations about the Dirichlet prior vector β. Note that each component of β, β_w, corresponds to a word w in the vocabulary $\mathcal{W}$. We found that a word w appears more frequently and more evenly over topics, it has larger chance to be associated with a higher β_w. For example, in DBLP, the word *algorithm* has higher β_w than the word *clustering*.

c) For a component of γ: For γ_a, the ath component of γ, based on Formulas (3) and (6), we have the conditional probability of γ_a, γ's ath component:

$$p(\gamma_a = x|\mathcal{Z}, \alpha, \beta, \gamma^{(-a)}, \mathcal{D}) \;\propto\; \frac{1}{x}\exp(-\frac{\ln^2 x}{2\sigma^2}) \prod_{t=1}^{T} \frac{\prod_{k=1}^{m_{ta}}(k - 1 + x)}{\prod_{k=1}^{m_t}(k - 1 + x + \sum_{i=1}^{A-1} \gamma_i^{(-a)})} \tag{11}$$

The formula above indicates that if an author name a appears more frequently and more evenly over topics, it has larger change to be associated with a higher γ_a. In DBLP, if an author name has more publications and the publications cover more topics, it has a higher γ_a.

C. The method to sample components of α, β and γ

The conditional distributions in Formulas (9), (10) and (11) are continuous probability density functions. Therefore, we employ the method of *Metropolis-Hastings within Gibbs* [17], [19]. We use log-normal distribution as the proposal density

$$Q(x'; x) \propto \frac{1}{x'} \exp\left(-\frac{(\ln x' - \ln x)^2}{2\sigma^2}\right),$$

from which a tentative new value x' are drawn, given the current value x. So we have $a' = \frac{P(x')}{P(x)}\frac{Q(x;x')}{Q(x';x)} = \frac{x'P(x')}{xP(x)}$ for the Metropolis-Hastings algorithm, where $P(x)$ is the right part of any one of Formula (9), (10) and (11). If $a' > 1$, the tentative new value is accepted; otherwise, it is accepted by probability a'. The proposal density's parameter σ is tuned to guarantee that the acceptance rate is around 0.5. This value is recommended for the random walk chain [17].

D. Model parameters after sampling

After Gibbs sampling is finished, we get the model parameters: α, β, γ and $\mathcal{Z}$. And we can estimate Φ, Ψ, and $p(t)$. The tth row and wth column of Φ is

$$\phi_{tw} = p(w|t) = \frac{n_{tw} + \beta_w}{n_t + \sum_{k=1}^{W} \beta_k}. \tag{12}$$

Here n_{tw} and n_t come from $\mathcal{Z}$. Formula (12) is an extension to the estimation of ϕ in [12] that is for single-value β. The

tth row and ath column of Ψ is

$$\psi_{ta} = p(a|t) = \frac{m_{ta} + \gamma_a}{m_t + \sum_{k=1}^{A} \gamma_k}. \tag{13}$$

The probability of a topic t is

$$p(t) = \frac{n_t + m_t + \alpha_t}{N + M + \sum_{k=1}^{T} \alpha_k}. \tag{14}$$

Here, N is the number of words and M is the number of author names in the corpus $\mathcal{D}$, and other symbols are the same as in Propositions 1, 2 and 3. Then we can apply the derived model parameters to a test document $\tilde{\mathbf{d}}$.

E. The illustrative example revisited

In the example in Section II-C, after learning the model by Gibbs sampling, model parameters could be as follows.
Dirichlet prior vectors:

$\alpha = (1.1, 0.9)$, $\beta = (0.4, 0.6, 1.1, 0.9)$, $\gamma = (0.9, 0.3, 0.6, 0.2)$.

Topic assignment for Document #1:

1.1 George Bush$^{(2)}$, D. Cheney$^{(1)}$: $Iraq^{(1)}$ $war^{(2)}$.

1.2 George Bush$^{(2)}$, C. Powell$^{(2)}$: $Gulf^{(2)}$ $war^{(1)}$.

1.3 George Bush$^{(1)}$: $Gulf^{(2)}$ $report^{(2)}$.

Topic assignment for Document #2:

2.1 George W. Bush$^{(1)}$, D. Cheney$^{(1)}$: $Iraq^{(1)}$ $report^{(1)}$.

Here superscripts $^{(1)}$ and $^{(2)}$ denote *samples of topic* for words and author names. For example, “George Bush$^{(2)}$” means topic 2 is assigned to the author name “George Bush”.

Based on the parameters given above, by Formula (12), we obtain the matrix Φ, the topics’ distributions over words in $\mathcal{W}$ (ϕ_i is for topic i):

	Gulf	Iraq	report	war
ϕ_1	0.06	0.37	0.30	0.27
ϕ_2	0.34	0.09	0.30	0.27

.

By Formula (13), we obtain the matrix Ψ, the topics’ distributions over author names in $\mathcal{A}$ (ψ_i is for topic i):

	George Bush	George W. Bush	D. Cheney	C. Powell
ψ_1	0.32	0.22	0.43	0.03
ψ_2	0.58	0.06	0.12	0.24

By Formula (14), we obtain topics’ probabilities: $p(t = 1) = 0.54$, $p(t = 2) = 0.46$.

F. Smoothing for new words or author names

If a test document is not in the corpus $\mathcal{D}$, it is possible that it contains new words that are not in the vocabulary $\mathcal{W}$. How to deal with the new words is an issue. When we try to represent a new document by a mixture of topics, using Formula (15), we need the value of ϕ_{tw} for each word. But for a new word, we cannot compute ϕ_{tw} by Formula (12) because of no corresponding β_w. Therefore, we use the following method for computing ϕ_{tw} for new words in a test document.

For each new word in test document $\tilde{\mathbf{d}}$, we append it to the vocabulary $\mathcal{W}$, and extend current W-dimensional β to a $(W+1)$-dimensional vector by adding a component with the value $min\{\beta_1, \ldots, \beta_W\}$, and then W is incremented by 1. We repeat this process until all new words have been added to $\mathcal{W}$. For each new word, we let its n_{tw} be 0.

On the other hand, there can be new author names in a test document. Similarly, we can update γ and $\mathcal{A}$ as we do to β and $\mathcal{W}$.

After new words and new author names are included in $\mathcal{W}$ and $\mathcal{A}$, Φ and Ψ can be re-computed by Formulas (12) and (13) and applied to the new document. By Formulas (15) and (16), the new document can be represented by a mixture of topics.

G. Document representation by model parameters

1) A document as a mixture of topics: Based on the derived model parameters, we can estimate a test document’s mixture of topics. We define the test document $\tilde{\mathbf{d}}$’s weight for a topic t is as below

$$W(\tilde{\mathbf{d}}, t) = p(t) \prod_{i=1}^{N_{\tilde{d}}} \phi_{tw_i}^{1/N_{\tilde{d}}} \prod_{j=1}^{M_{\tilde{d}}} \psi_{ta_j}^{1/M_{\tilde{d}}} \tag{15}$$

where $N_{\tilde{d}}$ and $M_{\tilde{d}}$ are the numbers of words and author names in $\tilde{\mathbf{d}}$, respectively; w_i and a_j as subscripts are the order numbers of $\tilde{\mathbf{d}}$’s ith word and jth author name in $\mathcal{W}$ and $\mathcal{A}$, respectively. Other notations have been defined in Formulas (12), (13) or (14).

Therefore we can get the document $\tilde{\mathbf{d}}$’s distribution over topics, denoted by the vector $\hat{\theta}_{\tilde{d}} = (\hat{\theta}_{\tilde{d}1}, \ldots, \hat{\theta}_{\tilde{d}T})$, where

$$\hat{\theta}_{\tilde{d}t} = p(t|\tilde{\mathbf{d}}) = \frac{W(\tilde{\mathbf{d}}, t)}{\sum_{t=1}^{T} W(\tilde{\mathbf{d}}, t)} \tag{16}$$

and $t = 1, 2, \ldots, T$.

2) Topic similarity of two documents: The topic similarity of two documents should be proportional to the probability that the two documents $\mathbf{d}_1$ and $\mathbf{d}_2$ belong to the same topic. Therefore, based on Formula (16), it is defined as

$$TopicSim(\mathbf{d}_1, \mathbf{d}_2) \propto \sum_{t=1}^{T} p(t) \hat{\theta}_{d_1 t} \hat{\theta}_{d_2 t}. \tag{17}$$

This formula is used in Section IV.

IV. Complete entity resolution

In this section, we present an application of the LDA-dual model — complete entity resolution for all author names in a given corpus.

Complete entity resolution consists of four steps: (1) obtain topic information for each document or citation in the corpus through Gibbs sampling; (2) build a classifier based on pairwise feature of two citations; (3) solve the name sharing problem by spectral clustering after constructing graphs for each ambiguous name, with the classifier’s support; (4) solve the name variant and name mixing problems with help of the classifier. The first step has been presented in Sections III and III-G. We now introduce the other three steps.

Based on model parameters, the topic similarity of two citations can be computed by Formula (17). Based on topic similarity and some other features, we build a classifier. Its features are a series of similarities or distances between two citations. Its class label has two values — *merge* when it decides that the two citations are from the same entity, and *not merge* otherwise.

To solve the name sharing problem, we use spectral clustering based on the classifier that has been learned. Given an ambiguous name shared by multiple entities, all citations that the name appears in can be obtained. For this name, we construct a graph in which citations are nodes. Edges of the graph are decided by the classifier. For a pair of nodes, an edge is assigned to them, only if their corresponding two citations are decided to be *merge* by the classifier.

If the classifier was perfect, i.e., free of errors, the graph would consist of some *cliques* and each clique would represent an entity. In this ideal case, it is trivial to find entities. In case the classifier has errors, K-way spectral clustering [23][24][25] is employed to retrieve entities after a graph is constructed.

After the name sharing problem is solved, we solve the name variant and name mixing problems based on the classifier previously learned. Given two names, we decide if they are from one entity. Two sets of citations are collected for the two names, respectively. Each pair of two citations from the two sets is decided by the classifier. If the ratio of *merge* surpasses a predefined threshold m_{thresh}, the two names are considered as from the same entity.

A. Building the classifier

The classifier is used to decide if two citations are from one entity. In our case, we tried the decision tree algorithm C4.5 and support vector machines.

We use the following features: (1) coauthor names similarity, (2) title similarity, (3) topic similarity, (4) venue similarity, and (5) minimum name distance between coauthors. Their values are computed based on a pair of citations. The values of the class label is *merge* and *not merge*. If two citations are from the same entity, we label it *merge*; otherwise, *not merge*.

Feature #3 topic similarity has been introduced in Section III-G by treating citations as test documents. We propose the notion *name distance* for features #1 and #5.

1) Name distance: We denote the *name distance* by δ. Consider two different author names a_1 and a_2: (1) If a_1 and a_2 have the same first name and last name and one of them has no middle name, we define $\delta(a_1, a_2) = 0.5$, e.g., *Clement Yu* and *Clement T. Yu*. (2) If a_1 and a_2 have the same first name and last name and one middle name is another middle name's initial, we define $\delta(a_1, a_2) = 0.5$, e.g., *Chee W. Chin* and *Chee Wee Chin*. (3) If a_1 and a_2 have the same last name and one first name is the other first name's initial, we define $\delta(a_1, a_2) = 0.5$, e.g., *Yang Zhang* and *Y. Zhang*. (4) If a_1 and a_2 have the same last name and the difference of two first names is only letter cases or '-', we define $\delta(a_1, a_2) = 0.5$, e.g., *Guohui Li* and *Guo-Hui Li*.

Except the scenarios above, the name distance of a_1 and a_2 is defined as their *edit distance*. The edit distance of two strings is defined as the minimum number of operations that transform one string to the other and the operations include inserting, deleting and replacing a character. Here, two author names are treated as two strings.

2) Coauthor names similarity: We use Dice's coefficient [26] to compute coauthor names similarity for two citations. Dice's coefficient is defined as $s = \frac{2|X \cap Y|}{|X|+|Y|}$, where X and Y are two sets. In this paper, considering similarity of similar names, we made a small improvement when applying Dice's coefficient. We add 0.5 to $|X \cap Y|$ in case the name distance is 0.5. For example, two sets of coauthor names are X = $\{$*Weiyi Meng, Clement T. Yu*$\}$ and Y = $\{$*Hongkun Zhao, Weiyi Meng, Clement Yu*$\}$. Considering similarity of *Clement T. Yu* and *Clement Yu*, we compute Dice's coefficient as $\frac{2\times1.5}{2+3} = 0.6$, while standard computation is $\frac{2\times1}{2+3} = 0.4$.

Feature #5 is the minimum name distance of two sets of coauthor names. For the example above, it is zero. Obviously, if the value of feature #5 is too large, the two citations are from different entities.

3) Title similarity: For two citations, the title similarity is computed as cosine similarity [26] based on *tf-idf* weights of words in the two titles. Each title is represented by a vector. Each component of the vector is computed based on term frequency (*tf*) and inverse document frequency (*idf*) of a word.

In addition, cosine similarity is also used to compute venue similarity of two citations.

4) Manual effort on labeling: We did not directly use unsupervised learning for citations under an author name to solve the name sharing problem. The reason is that it is difficult to determine the weights for all kinds of similarities. Instead, we employ a classifier.

To build the training set of the classifier, manual effort is needed to label each pair of citations as *merge* or *not merge*. In some cases, manual effort is little. For example, if we know there is only one entity under a name, a pair is labeled as *merge* easily if two citations in the pair are from this name. On the other hand, for two very different names, a pair can be labeled as *not merge* easily if one citation is from one name and the other citation from the other name.

B. Solving the name sharing problem

To solve the name sharing problem based on the classifier, we first construct a graph for each ambiguous name shared by K entities, and then employ K-way spectral clustering [23][24][25] to retrieve entities from this graph, as in Algorithm 1. K, the number of entities under an author name, is prior knowledge.

1) Constructing graphs: For a given ambiguous name, we first obtain a collection of all citations of the ambiguous name, and then construct a graph. Each node of the graph corresponds to a citation in the collection. The edges are determined by the classifier. If the classifier decides to *merge* the two citations, we draw an edge connecting two nodes corresponding to the two citations. Otherwise, no edge is drawn between them. Each edge has an equivalent weight that is 1.

2) Retrieving clusters: For K-way spectral clustering, there are three methods of spectral relaxation to apply — ratio cut [23], normalized cut [25] and min-max cut [24]. We applied the three methods to the application in this paper and found that ratio cut has the best performance.

Algorithm 1 solving the name sharing problem

```
Input:
$S$: classifier that decides two citations as *merge* or *not merge*,
$a$: an ambiguous author name,
$C$: the set of citations of author name $a$,
$k$: the number of entities sharing author name $a$.
Output:
$C_1, \ldots, C_k$: the sets of citations for $k$ identified entities, where $C_i \in C$ and $\bigcup_{i=1}^{k} C_i = C$.
Method:
Create a graph $G(V, E)$
$V(G) \leftarrow C$   $\{V$ is the set of vertices of graph $G\}$
$E(G) \leftarrow \phi$   $\{E$ is the set of edges of graph $G\}$
for each citation $i \in C$ do
    for each citation $j \in C$ and $j \neq i$ do
        if $S$ decides citation pair $(i, j)$ as "merge" then
            $E \leftarrow E \cup \{(v_i, v_j)\}$
            $\{(v_i, v_j)$ is an edge connecting vertices $v_i$ and $v_j\}$
$U \leftarrow [u_1, \ldots, u_k]$   $\{u_1, \ldots, u_k$ are eigenvectors corresponding to the $k$ smallest eigenvalues of the Laplacian matrix of graph $G\}$
Use $k$-means to cluster rows of $U$ into $k$ clusters.
Assign each citation in $C$ into $k$ clusters accordingly. $\{$each row of $U$ corresponds to a vertex in $G$ and a citation in $C\}$
return $C_1, \ldots, C_k$
```

According to Ky Fan's theorem [27], we can retrieve clusters from the subspace of k eigenvectors corresponding to the k smallest eigenvalues of the Laplacian matrix L of the graph, using another algorithm like k-means [25].

Assume we have obtained k eigenvectors of L: $u_1, \ldots, u_k$. Let $U = [u_1, \ldots, u_k]$. Each row of matrix U corresponds to a node in the graph. We cluster rows of U by k-means and then we obtain the clusters of nodes.

C. Solving the name variant problem

After the name sharing problem is solved, we solve the name variant problem by Algorithm 2.

Given two names a_1 and a_2, if the name distance $\delta(a_1, a_2)$ is larger than a predefined threshold d_{thresh}, say 3, we think they are from different entities. Otherwise we need to use the classifier to determine if they are from the same entity.

Two sets of citations, C_1 and C_2, are collected for a_1 and a_2, respectively. Any citation $c_1 \in C_1$ will be compared to any citation $c_2 \in C_2$. All pairs (c_1, c_2) will be decided to *merge* or *not merge* by the classifier. Let d be the count of *merge*. If the ratio of *merge* $\frac{d}{mn}$ is greater than a threshold m_{thresh}, say, 0.3, we consider the two names a_1 and a_2 are from the same entity; otherwise they are from different entities. We will discuss selection of m_{thresh} in Section V-F.

After the name sharing problem and the name variant problem are solved, the name mixing problem is also solved because it is a combination of the former two problems.

Algorithm 2 solving the name variant problem

```
Input:
$S$: classifier that decides two citations as *merge* or *not merge*,
$A$: a set of author names,
$d_{thresh}$: an admission threshold of name distance,
$m_{thresh}$: a threshold for name merging.
Output:
$P$: a set of pairs of merged author names.
Method:
$P \leftarrow \phi$
for each author name $a_1 \in A$ do
    $C_1 \leftarrow$ set of citations of $a_1$
    for each author name $a_2 \in A$ and $a_2 \neq a_1$ do
        if name distance $\delta(a_1, a_2) \leq d_{thresh}$ then
            $C_2 \leftarrow$ set of citations of $a_2$
            $numMerge \leftarrow 0$
            for each citation $i \in C_1$ do
                for each citation $j \in C_2$ do
                    if $S$ decides citation pair $(i, j)$ as "merge" then
                        $numMerge \leftarrow numMerge + 1$
            $mergeRatio \leftarrow \frac{numMerge}{|C_1||C_2|}$
            if $mergeRatio \geq m_{thresh}$ then
                $P \leftarrow P \cup \{(a_1, a_2)\}$
return $P$
```

V. Experimental results

A. Data sets

We choose the DBLP bibliography [1] as the test bed for our experimentation. Within the DBLP website, we can obtain a web page for an author name as shown in Figure 1. The web page lists all citations for which the author name appears as one of the coauthor. Each citation includes coauthors' names and links to their web pages, paper titles, venues of publication, publication years, etc.

A perfect corpus for experiments is the whole DBLP bibliography. For this case, Gibbs sampling process needs prohibitory hardware support like a main frame, or even a supercomputer that work in [12] was done on. Due to hardware limits, in our experiments, we select a medium-size corpus that is a subset of the whole bibliography.

To collect data for the corpus, we start from several author names' DBLP web pages and crawl for a few levels following the links on the web pages. Please note that there is no special criterion for the selection of the initial author names. The corpus includes 82721 words and 28678 author names, and the size of the vocabulary is 7736 and the size of author name list (the number of distinct author names) is 7470. Our objective is complete entity resolution in the corpus, assuming a list of ambiguous author names is given and the number of entities behind each ambiguous author name is known.

B. Gibbs sampling process

Because we include α, β and γ in the sampling process, our performance is not very sensitive to the selection of topics

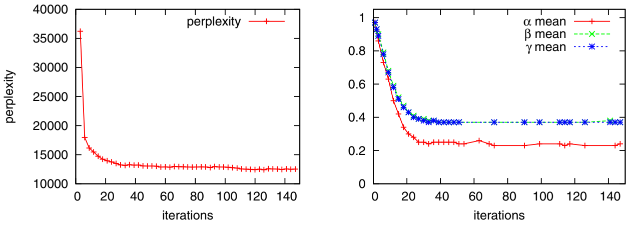


Fig. 4. As the number of iterations increases, the perplexity decreases (left) and the means of components of α, β and γ converge (right).

T. Generally, the larger the corpus is, the bigger T is. In our case, we select T to be 50. Furthermore, our approach is more adaptive or less sensitive to the selection of T, because the sampling will tune the values of α, β and γ in response to different T selections.

Another parameter is σ of log-normal distributions of each component of α, β and γ. We find when $1 \leq \sigma \leq 2$, the performance is the best.

One iteration of the Gibbs sampling process includes sampling for every word and author name in corpus $\mathcal{D}$ according to Formulas (7) and (8) and sampling for every component of vectors α, β and γ.

In our method, sampling of vectors α, β and γ has the same complexity as sampling of words and author names. So the time complexity of sampling was not increased due to adding vectors to the sampling process.

The sampling process is finished when the perplexity is stable. The perplexity is the reciprocal of geometric mean of the probabilities of words and names, similar as in [8]. The smaller it is, the better. Generally after 150 iterations, less than one hour, it becomes stable. Figure 4 shows that as the number iterations increases, the perplexity decreases (left) and α, β and γ converge (right).

In addition, once sampling is finished, the topic similarity of any two citations can be computed as in Section III-G.2.

C. Precision and recall

we adopt precision and recall as our measurement of performance. First we define two sets: $S_a = \{(i,j)|$ citations i and j are merged by algorithm a, and $i \neq j\}$, and $S_r = \{(i,j)|$ citations i and j are merged in reality, and $i \neq j\}$. Then we define precision and recall for algorithm a as follows:

$$precision = \frac{|S_a \cap S_r|}{|S_a|}, \quad recall = \frac{|S_a \cap S_r|}{|S_r|}.$$

F-measure is defined as the harmonic mean of precision and recall. This measure is directly applied to the classifier in Section IV-A and Algorithm 1. For Algorithm 2, we can redefine S_a and S_r by replacing citations with author names, and precision and recall can be calculated similarly.

D. Training the classifier

The classifier is used to determine if two citations are from the same entity. Its performance is critical because it is the basis of Algorithms 1 and 2. We compare two classifiers — the decision tree algorithm C4.5 and support vector machines (SVMs) in the Weka machine learning package [28].

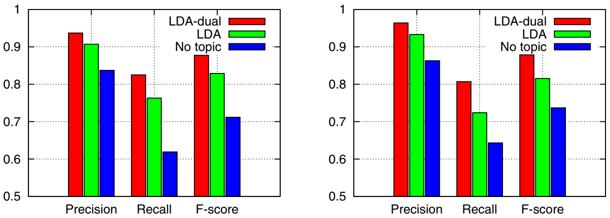


Fig. 5. The performance of classifiers C4.5 (left) and SVMs (right) on data sets for LDA-dual, LDA and no topic model, respectively.

To train the classifier, we use training data that is from about 2200 pairs of citations. The two citations in a pair are either from the citations of the same author name, or from similar author names. We eliminate the pairs of citations that share no similar coauthor names, because they are not useful when the classifier is used.

For each pair of citations, we compute values of five features: (1) coauthor name similarity, (2) title similarity, (3) topic similarity, (4) venue similarity, (5) minimum name distance between coauthors. Then the five values construct one instance. This instance's class label is prior knowledge, either "merge" (*positive*) or "not merge" (*negative*). For n pairs of citations, we have n instances with class label values. This is our training data set for the classifier.

Among the five features, feature #3 *topic similarity* is computed based on global information provided by the LDA-dual model described in Section II. This is an important improvement in this paper. To find out how useful this feature is, we perform comparison experiments on three different training data sets, each of which is based on one of the models as follows.

- LDA-dual model: feature #3 is computed based on the LDA-dual model and all five features are used in the data set.
- LDA model: feature #3 is computed based on the LDA model and all five features are used in the data set.
- No topic model: feature #3 is filtered out and only four features are used in the data set.

All training data sets have the same class label feature and the same number of instances.

We test classifiers by 10-fold cross validation and get average precision, recall and F-measure. The comparison results are in Figure 5. The classifiers have the better performance on topic models than no topic model. And among the topic models, the LDA-dual model is better than the LDA model. From this comparison, we can also see C4.5 and SVMs have almost the same performance.

E. Solving the name sharing problem

For an author name that is shared by multiple entities, we treat each citation under this name as a document. Based on the classifier, we construct a graph and do spectral clustering with three methods for spectral relaxations — ratio cut, normalized cut, and min-max cut. The ratio cut makes the number of vertices balanced in clusters, while the normalized cut makes the number of edges balanced. The min-max cut has no preference for balance. We compared three cuts on our data

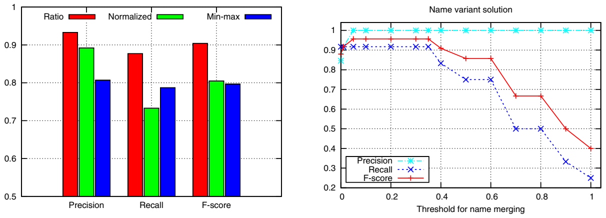


Fig. 6. (Left) Comparison of ratio cut, normalized cut and min-max cut of spectral clustering in solving the name sharing problem. The ratio cut is the best. (Right) The impact of *threshold for name merging* to performance of the name variant solution.

TABLE II

Part of result for the name sharing problem

name	*entities*	*citations*	*precision*	*recall*	*F-measure*
Michael Johnson	5	23	0.795	0.883	0.837
Tom Smith	3	6	1.0	1.0	1.0
William Smith	2	5	1.0	1.0	1.0
Huan Li	6	14	0.933	0.824	0.875
Guohui Li	4	20	0.973	0.947	0.959
Guo-Hui Li	2	4	1.0	1.0	1.0
Lei Yu	3	14	1.0	1.0	1.0
Hui Fang	3	9	1.0	1.0	1.0
Joseph Hellerstein	2	4	1.0	1.0	1.0

when we run Algorithm 1 and found that ratio cut has the best performance, as shown in Figure 6 (left).

The experimental result for some cases are listed in Table II. We have precision and recall over 0.9 for most tested cases. And in the cases of *Guohui Li* and *Lei Yu*, there are entities who moved from one place to another and our algorithm successfully identified them.

To minimize bias during learning, before solving the name sharing problem for an ambiguous author name, citations related to the author name are removed from the training data and the classifier is re-trained. And if the author name does not appear in the training data, the classifier is used without re-training.

F. Solving the name variant problem

We solve the name variant problem by Algorithm 2. There are two thresholds to be determined in advance: d_{thresh} and m_{thresh}. d_{thresh} is the admission threshold of name distance. If the name distance of two author names is greater than this value, they are considered from different entities and are dismissed. This threshold determines efficiency of the algorithm. If d_{thresh} is too large, the algorithm will be time consuming. In our experiment, we let d_{thresh} be between 1 and 3. The other threshold m_{thresh} is a threshold for name merging, and $0 \leq m_{thresh} \leq 1$. For a pair of author names, if their merging rate of citation pair surpasses m_{thresh}, they are considered from the same entity. Figure 6 (right) gives the relationship of m_{thresh} and precision, recall and F-measure of this algorithm. As m_{thresh} increases, precision increases and recall decreases. When $0.1 \leq m_{thresh} \leq 0.35$, the algorithm has the highest F-measure. Our algorithm has a high precision 0.99 and recall 0.917. This is because the classifier that has been learned earlier has high performance, after including topic information of the corpus. Previous work in [7] achieved precision 0.997 and recall 0.988 on the Citeseer data set. The performance difference could result from different data sets used. Another potential reason is that [7] considered only the name variant problem, while we consider the name sharing problem and the name mixing problem as well as the name variant problem. That makes our tasks and criteria a little different.

VI. Related Work

The related work include two facets — probabilistic models and entity resolution.

Probabilistic models

Probabilistic models of latent variables have been proposed and applied in information retrieval for years. [29] introduced probabilistic Latent Semantic Indexing (pLSI), an approach for automated document indexing based on a statistical latent class model. [8] pointed out a limitation of the pLSI model – providing no probabilistic model at the level of document, and then proposed Latent Dirichlet Allocation (LDA), a generative probabilistic model, with the latent variable *topic* introduced. Based on the LDA model, [12] employed Gibbs Sampling, a Monte Carlo method [16][19][20][18], to discover topics in a corpus. And the number of topics is estimated by maximizing log likelihood. [30][31] extended LDA by introducing *author* into the model. In the model documents are generated by topic-word and author-topic distributions that are learned from data. But it is not clear that the model can be used in name disambiguation. [7] extended and applied LDA to entity resolution. The document, topic and word previously in LDA correspond to the citation, group label and author reference, respectively, in [7]. The group label is a latent variable already. And they added another latent variable – author label (unambiguous author name) and a noise model of name variants.

Entity resolution

There have been quite a few papers on entity resolution or name disambiguation. Some focus on *name sharing* problem [2][5][3][6][4]. [3] aimed to solve the name sharing problem by constructing and analyzing a social network. [6] tried to solve the name sharing problem by analyzing results returned from search engines. [4] proposed DISTINCT method. Some other researchers aimed at solving the *name variant* problem, like [7][32][33][34]. The problem is also called entity resolution.

VII. Conclusions

In this paper, we proposed a new solution to the problem of complete entity resolution.

The LDA-dual model is a graphical model proposed as an extension to the LDA model by considering two types of information in documents, while the LDA model considers only one type of information. From the LDA-dual model we derived a corpus's posterior distribution and three propositions. We then proposed an improved process to learn the model from a corpus by Gibbs sampling, including Dirichlet priors in the sampling. The model is then applied to help solve two subproblems of complete entity resolution.

One main difference between our approach and other techniques is that we emphasize the use of the global information of a corpus like topics based on learning the LDA-dual model.

By this global information, we learn a high-performance classifier on which our complete entity resolution is based.

We have carried out preliminary experiments to evaluate the accuracy of the proposed solution. The results are very promising. For the cases we tested, both the *name sharing* and the *name variant* problems were solved with high accuracy. We plan to test our solution using much larger data sets in the near future, and further apply the LDA-dual model to other domains with two types of information for document analysis or other applications. The current smoothing method for new words and new author names does not scale, and it will be improved in the future.

Acknowledgment

This work is supported in part by the following NSF grants: IIS-0414981 and CNS-0454298.

References

[1] M. Ley, *DBLP Computer Science Bibliography*. http://www.informatik.uni-trier.de/∼ley/db/.

[2] H. Han, H. Zha, and L. Giles, "Name disambiguation in author citations using a k-way spectral clustering method," in *JCDL*, 2005.

[3] B. Malin, "Unsupervised name disambiguation via social network similarity," in *SIAM*, 2005.

[4] X. Yin, J. Han, and P. S. Yu, "Object distinction: Distinguishing objects with identical names," in *ICDE*, 2007.

[5] D. Lee, B.-W. On, J. Kang, and S. Park, "Effective and scalable solutions for mixed and split citation problems in digital libraries," in *Information Quality in Informational Systems*.

[6] Y. F. Tan, M.-Y. Kan, and D. Lee, "Search engine driven author disambiguation," in *JCDL*, June 2006.

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

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[10] T. M. Mitchell, *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997.

[11] E. Artin, *The Gamma Function*. New York: Holt, Rinehart, and Winston, 1964.

[12] T. L. Griffiths and M. Steyvers, "Finding scientific topics," in *Proceedings of the National Academy of Sciences*, 2004.

[13] C. Andrieu, N. D. Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine Learning*, vol. 50, pp. 5–43, 2003.

[14] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "Introduction to variational methods for graphical models," *Machine Learning*, vol. 37, pp. 183–233, May 1999.

[15] T. Minka and J. Lafferty, "Expectation-propagation for the generative aspect model," in *18th Conference on Uncertainty in Artificial Intelligence*, pp. 352–359.

[16] K. Binder, "Applications of monte carlo methods to statistical physics," *Rep. Prog. Phys.*, no. 60, pp. 487–559, 1997.

[17] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, November 1995.

[18] A. E. Gelfand, "Gibbs sampling," *Journal of the American Statistical Association*, vol. 95, no. 452, pp. 1300–1304, December 2000.

[19] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[20] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, June 2000.

[21] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[22] E. Limpert, W. Stahel, and M. Abbt, "Log-normal distribution across the sciences: Keys and clues," *BioScience*, vol. 51, no. 5, pp. 341–352, May 2001.

[23] P. Chan, M. Schlag, and J. Zien, "Spectral k-way ratio-cut partitioning and clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 9, pp. 1088–1096, September 1994.

[24] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *ICDM*, 2001.

[25] A. Y. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NIPS 14*, 2002.

[26] A. R. Webb, *Statistical Pattern Recognition (2nd Edition)*. John Wiley and Sons Ltd., 2002.

[27] R. Bhatia, *Matrix analysis*. New York: Springeer-Cerlag, 1997.

[28] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques, Second Edition*. San Francisco: Morgan Kaufmann, 2005.

[29] T. Hofmann, "Probabilistic latent semantic indexing," in *ACM SIGIR Conference*, 1999.

[30] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *UAI*, 2004.

[31] M. Rosen-Zvi, T. Griffiths, P. Smyth, and M. Steyvers, "Learning author topic models from text corpora," in *SIGKDD*, 2005.

[32] B.-W. On, E. Elmacioglu, D. Lee, J. Kang, and J. Pei, "Improving grouped-entity resolution using quasi-cliques," in *ICDM*, 2006.

[33] P. Reuther, "Personal name matching: New test collections and a social network based approach," *Tech. Report, Univ. Trier*, 2006.

[34] J. Xia, "Personal name identification in the practice of digital repositories," *Program*, vol. 40, no. 3, pp. 256–267, 2006.

Appendix

A. Proof of Proposition 1

Proof: Because of the Dirichlet distribution of θ_d, the dth row vector of Θ, we have $p(\theta_d|\alpha) = \frac{\Gamma(\sum_t \alpha_t)}{\prod_t \Gamma(\alpha_t)} \prod_t \theta_{dt}^{\alpha_t - 1}$, where α_t and θ_{dt} are the tth component of α and θ_d, respectively, and $\Gamma(\cdot)$ is the Gamma function [11]. Also considering the multinomial distributions of $z^{(w)}$ and $z^{(a)}$, we have

$$p(\mathcal{Z}, \Theta|\alpha) = p(\Theta|\alpha) p(\mathcal{Z}^{(w)}|\Theta) p(\mathcal{Z}^{(a)}|\Theta) =$$

$$\prod_{d=1}^{D} \left\{ p(\theta_d|\alpha) \left[\prod_{n=1}^{N_d} p(z_{dn}^{(w)}|\theta_d) \right] \left[\prod_{m=1}^{M_d} p(z_{dm}^{(a)}|\theta_d) \right] \right\} =$$

$$\prod_{d=1}^{D} \left\{ \left[\frac{\Gamma(\sum_{t=1}^{T} \alpha_t)}{\prod_{t=1}^{T} \Gamma(\alpha_t)} \prod_{t=1}^{T} \theta_{dt}^{\alpha_t - 1} \right] \left[\prod_{t=1}^{T} \theta_{dt}^{n_{dt}} \right] \left[\prod_{t=1}^{T} \theta_{dt}^{m_{dt}} \right] \right\} =$$

$$\prod_{d=1}^{D} \left\{ \frac{\Gamma(\sum_{t=1}^{T} \alpha_t)}{\prod_{t=1}^{T} \Gamma(\alpha_t)} \prod_{t=1}^{T} \theta_{dt}^{\alpha_t + n_{dt} + m_{dt} - 1} \right\}.$$

Using $\Gamma(x+1) = x\Gamma(x)$ repeatedly, we can derive $\Gamma(n+x) = \Gamma(x) \prod_{k=1}^{n} (k - 1 + x)$. Integrating over Θ and considering $\sum_{t=1}^{T} (n_{dt} + m_{dt}) = N_d + M_d$, we have

$$p(\mathcal{Z}|\alpha) = \int p(\mathcal{Z}, \Theta|\alpha) d\Theta =$$

$$\prod_{d=1}^{D} \left\{ \frac{\Gamma(\sum_{t=1}^{T} \alpha_t)}{\prod_{t=1}^{T} \Gamma(\alpha_t)} \cdot \frac{\prod_{t=1}^{T} \Gamma(n_{dt} + m_{dt} + \alpha_t)}{\Gamma(N_d + M_d + \sum_{t=1}^{T} \alpha_t)} \right\} =$$

$$\prod_{d=1}^{D} \frac{\prod_{t=1}^{T} \prod_{k=1}^{n_{dt} + m_{dt}} (k - 1 + \alpha_t)}{\prod_{k=1}^{N_d + M_d} (k - 1 + \sum_{t=1}^{T} \alpha_t)}.$$

This completes the proof of Proposition 1. ■