A Combination Approach to Web User Profiling

JIE TANG
Tsinghua University
LIMIN YAO
University of Massachusetts Amherst
DUO ZHANG
University of Illinois at Urbana-Champaign
and
JING ZHANG
Tsinghua University

In this article, we study the problem of Web user profiling, which is aimed at finding, extracting, and fusing the "semantic"-based user profile from the Web. Previously, Web user profiling was often undertaken by creating a list of keywords for the user, which is (sometimes even highly) insufficient for main applications. This article formalizes the profiling problem as several subtasks: profile extraction, profile integration, and user interest discovery. We propose a combination approach to deal with the profiling tasks. Specifically, we employ a classification model to identify relevant documents for a user from the Web and propose a Tree-Structured Conditional Random Fields (TCRF) to extract the profile information from the identified documents; we propose a unified probabilistic model to deal with the name ambiguity problem (several users with the same name) when integrating the profile information extracted from different sources; finally, we use a probabilistic topic model to model the extracted user profiles, and construct the user interest model. Experimental results on an online system show that the combination approach to different profiling tasks clearly outperforms several baseline methods. The extracted profiles have been applied to expert finding, an important application on the Web. Experiments show that

This work was done when L. Yao and D. Zhang were studying in Tsinghua University.
This work is supported by the Natural Science Foundation of China (No. 60703059, No. 60973102), Chinese National Key Foundation Research (No. 60933013), National High-Tech R&D Program (No. 2009AA01Z138), and Chinese Young Faculty Research Fund (No. 20070003093).

Authors' addresses: J. Tang (corresponding author) and J. Zhang, Room 1-308, FIT Building, Tsinghua University, Beijing, 100084, China; email: jietang@tsinghua.edu.cn, zhangjing@keg.cs.tsinghua.edu.cn; D. Zhang, 1125 Siebel Center for Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin Ave., Urbana, IL 61801; email: dzhang22@illinois.edu; L. Yao, Department of Computer Science, University of Massachusetts Amherst, 140 Governors Drive, Amherst, MA 01003; email: lmyao@cs.umass.edu.

the accuracy of expert finding can be improved (ranging from +6% to +26% in terms of MAP) by taking advantage of the profiles.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.4.0 [**Information Systems Applications**]: General; H.2.8 [**Database Management**]: Database Applications

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: User profiling, information extraction, name disambiguation, topic modeling, social network, text mining

ACM Reference Format:
Tang, J., Yao, L., Zhang, D., and Zhang, J. 2010. A combination approach to Web user profiling. ACM Trans. Knowl. Discov. Data 5, 1, Article 2 (December 2010), 44 pages.
DOI: 10.1145/1870096.1870098. http://doi.acm.org/10.1145/1870096.1870098.

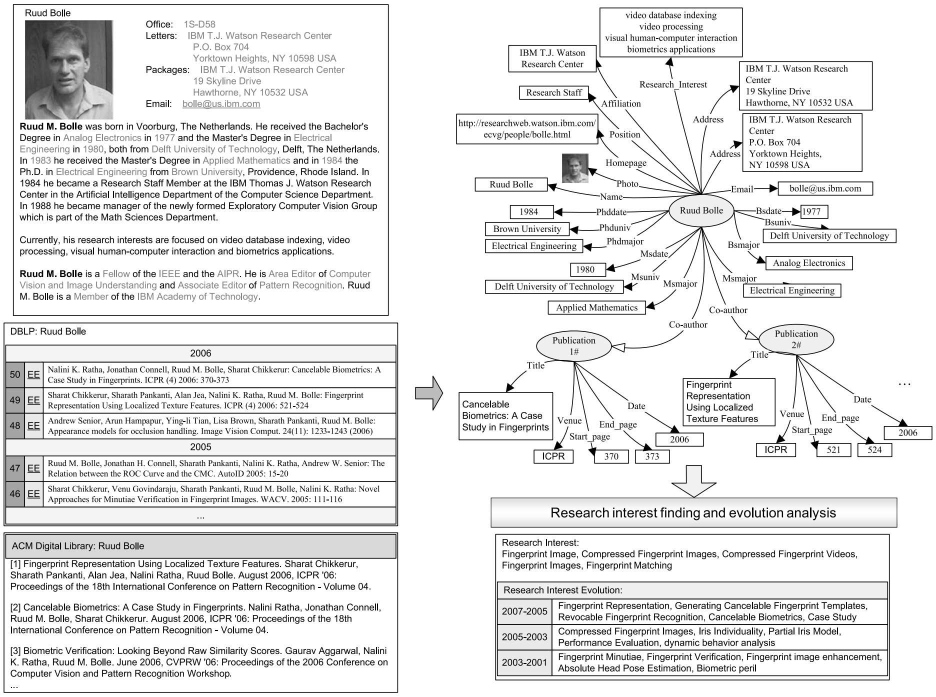
1. INTRODUCTION

Profiling of a Web user is the process of obtaining values of different properties that constitute the user model. Considerable efforts have been made to mine the user's interests from his/her historical data. A typical way for representing the user's interests is to create a list of relevant keywords. However, such a profile is *insufficient* for modeling and understanding users' behaviors. A complete user profile (including one's education, experience, and contact information) is very important for providing high-quality Web services. For example, with a well-organized user profile base, online advertising can be more targeted based on not only on a user's interests but also on his/her current position.

Traditionally, user profiling was viewed as an engineering issue and was conducted manually or undertaken separately in a more or less ad hoc manner. For instance, in Web-based social networks such as MySpace and YouTube, the user has to enter the profile by herself/himself. Unfortunately, the information obtained solely from the user entering a profile is sometimes incomplete or inconsistent. Users do not fill in some information merely because they are not willing to fill it in.

Some other work builds the user profile with A list of keywords generated using statistical methods, for example, using high-frequency words discovered from the user-entered information or user-browsed Web pages. However, such a method ignores some important semantic information such as location and affiliation.

Recently, some work has been conducted to automatically build the semantic-based user profile using information extraction technologies [Alani et al. 2003; Pazzani and Billsus 1997; Yu et al. 2005]. Most of the existing methods use predefined rules or specific machine learning models to extract the different types of profile information in a separated fashion. However, some profile information (e.g., user interests) is implied in the user-related documents (e.g., blogs) and cannot be explicitly extracted from the Web page.



Ruud Bolle

Office: 1S-D58
Letters: IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598 USA
Packages: IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532 USA
Email: bolle@us.ibm.com

Ruud M. Bolle was born in Voorburg, The Netherlands. He received the Bachelor's Degree in Analog Electronics in 1977 and the Master's Degree in Electrical Engineering in 1980, both from Delft University of Technology, Delft, The Netherlands. In 1983 he received the Master's Degree in Applied Mathematics and in 1984 the Ph.D. in Electrical Engineering from Brown University, Providence, Rhode Island. In 1984 he became a Research Staff Member at the IBM Thomas J. Watson Research Center in the Artificial Intelligence Department of the Computer Science Department. In 1988 he became manager of the newly formed Exploratory Computer Vision Group which is part of the Math Sciences Department.

Currently, his research interests are focused on video database indexing, video processing, visual human-computer interaction and biometrics applications.

Ruud M. Bolle is a Fellow of the IEEE and the AIPR. He is Area Editor of Computer Vision and Image Understanding and Associate Editor of Pattern Recognition. Ruud M. Bolle is a Member of the IBM Academy of Technology.

DBLP: Ruud Bolle

2006		
50	EE	Nalini K. Ratha, Jonathan Connell, Ruud M. Bolle, Sharat Chikkerur: Cancelable Biometrics: A Case Study in Fingerprints. ICPR (4) 2006: 370-373
49	EE	Sharat Chikkerur, Sharath Pankanti, Alan Jea, Nalini K. Ratha, Ruud M. Bolle: Fingerprint Representation Using Localized Texture Features. ICPR (4) 2006: 521-524
48	EE	Andrew Senior, Arun Hampapur, Ying-li Tian, Lisa Brown, Sharath Pankanti, Ruud M. Bolle: Appearance models for occlusion handling. Image Vision Comput. 24(11): 1233-1243 (2006)
2005		
47	EE	Ruud M. Bolle, Jonathan H. Connell, Sharath Pankanti, Nalini K. Ratha, Andrew W. Senior: The Relation between the ROC Curve and the CMC. AutoID 2005: 15-20
46	EE	Sharat Chikkerur, Venu Govindaraju, Sharath Pankanti, Ruud M. Bolle, Nalini K. Ratha: Novel Approaches for Minutiae Verification in Fingerprint Images. WACV. 2005: 111-116
...		

ACM Digital Library: Ruud Bolle

[1] Fingerprint Representation Using Localized Texture Features. Sharat Chikkerur, Sharath Pankanti, Alan Jea, Nalini Ratha, Ruud Bolle. August 2006, ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition - Volume 04.

[2] Cancelable Biometrics: A Case Study in Fingerprints. Nalini Ratha, Jonathan Connell, Ruud M. Bolle, Sharat Chikkerur. August 2006, ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition - Volume 04.

[3] Biometric Verification: Looking Beyond Raw Similarity Scores. Gaurav Aggarwal, Nalini K. Ratha, Ruud M. Bolle. June 2006, CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop.
...

Research Interest:
Fingerprint Image, Compressed Fingerprint Images, Compressed Fingerprint Videos, Fingerprint Images, Fingerprint Matching

Research Interest Evolution:	
2007-2005	Fingerprint Representation, Generating Cancelable Fingerprint Templates, Revocable Fingerprint Recognition, Cancelable Biometrics, Case Study
2005-2003	Compressed Fingerprint Images, Iris Individuality, Partial Iris Model, Performance Evaluation, dynamic behavior analysis
2003-2001	Fingerprint Minutiae, Fingerprint Verification, Fingerprint image enhancement, Absolute Head Pose Estimation, Biometric peril

Fig. 1. An example of researcher profiling.

1.1 Motivating Example

To clearly motivate this work, we demonstrate with an example drawn from a real-world system, ArnetMiner.[1] In this system, one basic goal is to create a profile for each researcher, which contains basic information (e.g., photo, affiliation, and position), contact information (e.g., address, email, and telephone number), educational history (e.g., university graduated from and major), research interests, and publications. For each researcher, some of the profile information can be extracted from his/her homepage or Web pages introducing him/her; some other profile information (e.g., publications) can be integrated from online digital libraries (e.g., DBLP or ACM); and the other information (e.g., research interests) can be mined from the collected information.

Figure 1 shows an example of a researcher profile. The left part shows the researcher's homepage and his DBLP/ACM page which contains his publication papers. The ideal profiling results are shown in the right part of Figure 1. The right-bottom part shows the researcher's interests mined from the publication papers.

Such a profiling result can benefit many data mining and social network applications. For example, if all researchers' profiles are correctly created, we will have a large collection of information making up a well-structured database about researchers around the world. We can use the profiles to help with

[1] http://www.arnetminer.org/

mining applications such as expert finding, which aims to find experts on a given topic.

The challenges of user profiling are as follows: (1) how to identify relevant pages for a given user and how to extract the profile information from the identified pages; (2) how to integrate the profiles extracted from different sources/pages, as the profile information of a user might be distributed on multiple pages; (3) how to discover user interests implied in the user associated documents.

Manually entering the user profile is obviously tedious and time consuming. Recent work has shown the feasibility and promise of information extraction technologies for extracting the structured data from the Web, and it is possible to use the methods to extract the profile of a user. However, most of existing methods employ a predefined rule or a specific machine learning model to separately identify each property of the profile. However, it is *highly ineffective* to use the separated methods to do profile extraction due to the natural disadvantages of the methods. (1) For each property in the profile, one has to define a specific rule or a specific supervised learning model. Therefore, there may be many different rules/models, which are difficult to maintain. (2) The separated rules/models cannot take advantage of dependencies across different properties. The properties are often dependent on each other. For instance, in Figure 1, identifying the text Electrical Engineering as *Msmajor* will greatly increase the probability of the text Delft University of Technology being identified as *Msuniv*. Consequently, how to effectively identify the profile information from the Web becomes a challenging issue.

For integration of the profile extracted from different sources, we need to deal with the name ambiguity problem (several users with the same name). Existing methods include heuristic rules, the classification-based supervised method, and the clustering-based unsupervised method. However, it is ineffective to directly employ the existing methods in user profile integration. This is because (1) the heuristic rule-based method requires the user to define a specific rule for each specific type of ambiguity problem, which is not adaptive for different situations; (2) the supervised method trains a user-dependent model for a certain person and thus cannot be adapted to other persons; and (3) the clustering-based unsupervised method cannot use the dependencies between papers and also cannot use the supervised information.

For the discovery of user interests, it is also insufficient to use the existing keyword-based methods. There are two main reasons: (1) these methods do not consider the semantic relationship between words, and (2) the methods ignore the dependencies between users, for example, users who coauthor many papers may have the same interests.

1.2 Our Solution

In this article, we aim to conduct a systematic investigation of the problem of Web user profiling. First, we decompose Web user profiling as three subtasks: profile extraction, name disambiguation, and user interest discovery. All of the three subtasks can be formalized using graphical models. Specifically, for profile extraction, as the information on the Web is naturally laid out in a hierarchical structure, we propose formalizing the problem in a tree-structured conditional random field. For name disambiguation, the problem is to assign papers to different persons with a same name. We formalize the problem in a Markov random graph, where each node denotes a paper and each edge denotes relationship (e.g., coauthor) between papers. For user interest discovery, we propose a generative graphical model, where the paper writing procedure is formalized in a series of probabilistic steps. To the best of our knowledge, our work is the first to formalize all the subtasks of user profiling in a combination approach and tackle all the problems at once.

We have implemented the proposed approaches in the system ArnetMiner.org. The system has been in operation on the Internet for more than 3 years and has attracted user accesses from 190 countries. In total, more than half a million researchers' profiles have been extracted. We have conducted experiments for extracting researchers' profiles. Experimental results indicate that our method clearly outperforms the methods of using separated models for profile extraction. Experimental results also indicate that our disambiguation method can outperform existing methods. We apply the proposed methods to expert finding. Experimental results show that our methods of profile extraction, name disambiguation, and user interest analysis can indeed enhance expert finding (+26% in terms of MAP).

Our contributions in this article include (1) a formalization of the problem of user profiling, (2) a proposal of a unified tagging approach to extract user profile, (3) a proposal of a probabilistic method to name disambiguation, (4) a proposal of a topic model to perform topical analysis of user interests, and (5) an empirical verification of the effectiveness of the proposed approaches. The approaches proposed in this article are general and can be applied to many applications, for example, social network extraction and information integration.

The rest of the article is organized as follows. In Section 2, we formalize the problem of Web user profiling. In Section 3, we give an overview of our approach. In Section 4, we explain our approach to profile extraction and in Section 5 we describe how we deal with the name ambiguity problem when integrating the extracted profiles. In Section 6, we present our method for user interests discovery. Section 7 gives the experimental results. Section 8 describes a demonstration system. Finally, before concluding the article in Section 10, we introduce related work.

2. PROBLEM FORMULATION

In different applications, definitions of profile schemas might be different. In this article, we use the researcher profile as the example for explanation. The definition of the researcher profile and the proposed approaches for user profiling can be easily extended to other applications.

We define the schema of the researcher profile (as shown in Figure 2) by extending the FOAF ontology [Brickley and Miller 2004]. In the schema, four concepts, 29 properties, and four relations are defined. The social network

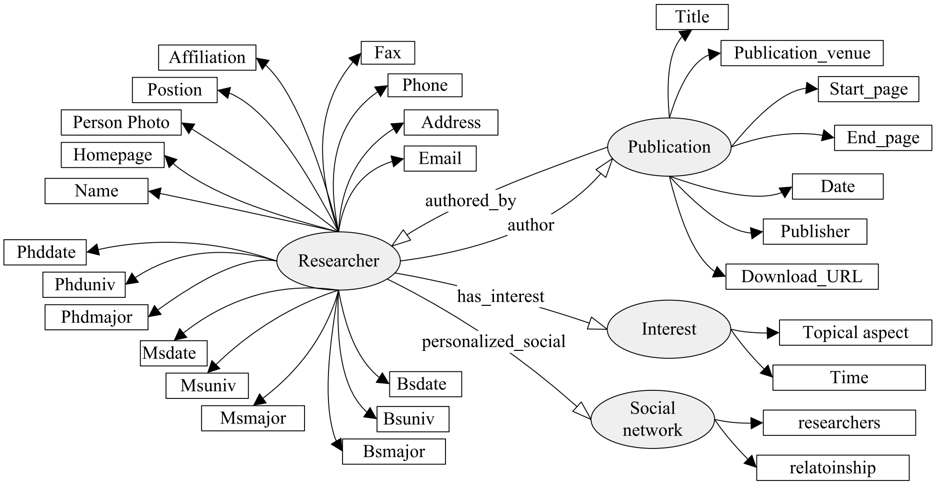


Fig. 2. Schema of researcher profile.

denotes the subsocial graph related to the current researcher. The interest denotes the semantic topical aspect, which will be detailed later. The publication denotes documents coauthored by the researcher.

We use the data from the ArnetMiner system for study. The system tries to provide a social networking platform for academic researchers. It has gathered 648,289 researcher profiles. Our statistical study shows that about 70.60% of the researchers have at least one homepage or a Web page that introduces them, which implies that extraction of the profile from the Web is feasible. For the name ambiguity problem (different researchers with the same name), we have examined 100 randomly selected researcher names and found that more than 30% of the names have the ambiguity problem.

We describe here the three key issues we are going to deal with: profile extraction, name disambiguation, and user interests.

(1) *Profile extraction.* We produced statistics on randomly selected 1,000 researchers. We observed that 85.6% of the researchers are faculties of universities and 14.4% are from company research centers. Researchers from the same company often have a template-based homepage. However, different companies have absolutely different templates. For researchers from universities, the layout and the content of the homepages varies largely depending on the authors. We have also found that 71.9% of the 1000 Web pages are researchers' homepages and the rest are pages introducing the researchers. Characteristics of the two types of pages significantly differ from each other.

We analyzed the content of the Web pages and found that about 40% of the profile properties are presented in tables or lists and the others are presented in natural language. This means a method without using the global context information in the page would be ineffective. Statistical study also unveils that (strong) dependencies exist between different profile properties. For example, there are $1,325$ cases (14.5%) in our data that property labels of the tokens need to use the extraction results of the other tokens. An ideal method should consider processing all the subtasks together.

Moreover, different from previous data extraction work, information on the Web page is usually organized hierarchically. For example, in the researcher homepage shown in Figure 1, the top information block contains the basic information (e.g., a photo, two addresses, and an email address), the middle block describes educational history information (e.g., universities graduated from and majors), and the bottom block includes the professional services information (e.g., position and affiliation information). An immediate observation is that identification of the type of the information block would be greatly helpful in identifying the information contained in the block.

(2) *Name disambiguation.* We do not perform extraction of publications directly from homepages. Instead, we integrate the publication data from existing online data sources. We chose DBLP bibliography,[2] which is one of the best formatted and organized bibliography datasets. DBLP covers approximately 1,200,000 papers from major computer science publication venues. In DBLP, authors are identified by their names. For integrating the researcher profiles and the publications data, we use researcher names and the author names as the identifiers. The method inevitably has the name ambiguity problem.

We give a formal definition of the name disambiguation task in our context. Given a person's name a, we denote all publications having the author name a as $P = \{p_1, p_2, \cdots, p_n\}$. For u authors of a paper $\{a_i^{(0)}, a_i^{(1)}, \cdots, a_i^{(u)}\}$, we call the author name we are going to disambiguate the *principal author* (denoted as $a_i^{(0)}$) and the others *secondary authors*. Suppose there are k actual researchers having the name a; our task is then to assign papers with the name a to their actual researcher y_h, $h \in [1, k]$.

(3) *User interests.* We do not extract research interests directly from the researchers' homepages, although we could do it in principle. There are two reasons: first, we observed only one-fifth (21.3%) of researchers provide their research interests on the homepages; second, research interests are usually implied by the associated documents, for example, papers published by the researcher.

Formally, we define user interest on the basis of topics. Each topic is defined as $z = \{(w_1, p(w_1|z)), \cdots, (w_{N1}, p(w_{N1}|z))\}$. The definition means that a topic is represented by a mixture of words and their probabilities belonging to the topic. The topic definition can be also extended to other information sources. For example, in the academic application, we can extend the topic definition by publication venues c, that is, $z = \{(c_1, p(c_1|z)), \cdots, (c_{N1}, p(c_{N1}|z))\}$. Finally, the interests of researcher a are defined as a set of topic distributions $\{P(z|a)\}_z$.

[2] dblp.uni-trier.de/

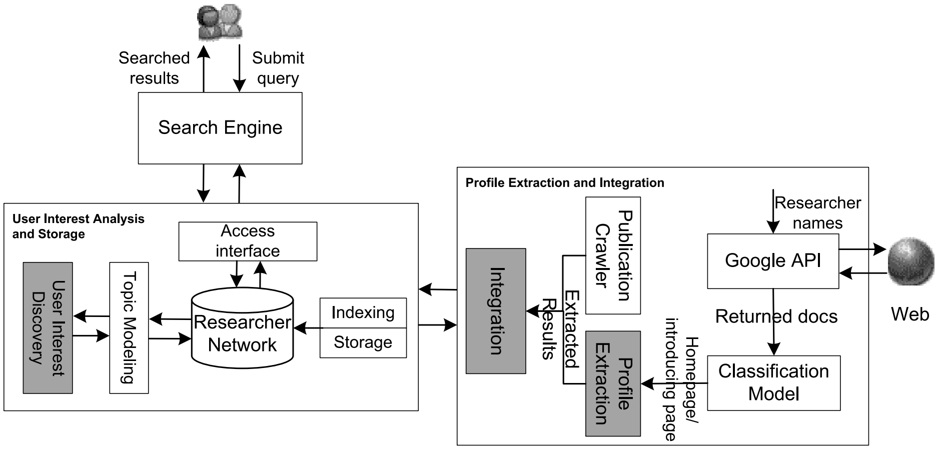


Fig. 3. Approach overview.

3. OVERVIEW OF OUR APPROACH

We propose a combination approach to solve the user profiling problem. Figure 3 shows the overview of our approach. There are mainly two components: profile extraction and integration, and user interest analysis. The first component targets extracting and integrating profile information from the Web; the second targets analyzing users' interests.

In the profile extraction and integration component, given a researcher name, we first use the Google API to retrieve a list of documents that contain that name. Then we employ a classification model to identify whether a document in the list is the homepage or an introducing page of the researcher. Next, we use an extraction model to extract the profile information from the identified pages. In particular, we view the problem as that of assigning tags to the input texts, with each tag representing a profile property.

We crawl the publication information from several online digital libraries (e.g., DBLP). We integrate the publication information and extracted profile information. We propose a probabilistic model to deal with the name ambiguity problem for integrating the extracted user profiles. The model can incorporate any type of domain background knowledge or supervised information (e.g., user feedbacks) as features to improve the performance of disambiguation.

In the user interest analysis component, we use a probabilistic topic model to discover the latent topic distribution associated with each researcher. Then we use the discovered topic distributions as the researcher interests.

In this article, our main technical contributions lie in the approaches we propose to deal with the three subtasks in the two components: profile extraction, integration, and user interest discovery. Theoretically, all three approaches are based on a probabilistic graphical model. More specifically, for profile extraction and integration, our approaches are based on the theory of Markov Random Field [Hammersley and Clifford 1971]. Markov Random Field (MRF) is a probability distribution of labels (hidden variables) that obeys the Markov property. It can be formally defined as follows.

Definition 3.1 *MRF Definition*. Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a Markov random field in the case where the random variable Y_v obeys the Markov property with respect to the graph: $p(Y_v|Y_w, w \neq v) = p(Y_v|Y_w, w \backsim v)$, where $w \backsim v$ means that w and v are neighbors in G.

The proposed model for profile extraction is a Tree-structured Conditional Random Fields (TCRFs) and the proposed model for name disambiguation is based on Hidden Markov Random Fields (HMRFs) [Basu et al. 2004]. The reasons we use the two models are as follows: (1) such models can describe the dependencies between information, and thus can improve the accuracy of profile extraction and name disambiguation. (2) For profile extraction, we can label some training data for supervised learning; while for name disambiguation it is difficult to provide sufficient training data. Therefore, we propose using a discriminative model (TCRF) for profile extraction and a generative model (HMRF) for the name disambiguation task. (3) Both models can be easily extended; thus for different applications we can extend the models based on application-specific features.

As for user interest analysis, the proposed model is a multilevel Bayesian network, which models each paper by following a stochastic process: first one of the paper's authors would decide what topic z to write according to his/her research interest (i.e., topic distribution) $\{P(z|a)\}_z$. Then a word w_{di} is sampled from the topic z according to the word distribution of the topic $\{P(w|z)\}_w$. This series of probabilistic steps can capture well the process of authors writing a paper. In addition, parameters (topic distribution and word distribution) can be estimated in an unsupervised way. Another reason for using the Bayesian network for user interest analysis is that we can easily incorporate different types of objects (e.g., researchers, publication venues, and papers) into one model; thus we can uncover the latent dependencies between the heterogeneous objects.

In the following sections, we will describe the proposed approaches in more detail.

4. PROFILE EXTRACTION

4.1 Process

There are three steps: relevant page finding, preprocessing, and tagging. In relevant page finding, given a researcher name, we first get a list of Web pages by a search engine (i.e., Google) and then identify the homepage or introducing page using a binary classifier. We use support vector machines (SVMs) [Cortes and Vapnik 1995] as the classification model and define features such as whether the title of the page contains the person name and whether the URL address (partly) contains the person name. The performance of the classifier is 92.39% by the $F1$ measure.

In preprocessing, (a) we segment the text into tokens and (b) we assign possible tags to each token. The tokens form the basic units and the pages form the sequences of units or a tree structure of units in the tagging problem. In tagging, given a sequence of units or a tree structure of units, we determine the most likely corresponding tags using a trained tagging model. Each tag corresponds to a property defined in Figure 2. In this article, we present a Tree-structure Conditional Random Fields (TCRF) [Tang et al. 2006] as the tagging model. Next we describe steps (a) and (b) in detail.

(a) We identify tokens in the Web page heuristically. We define five types of tokens: *standard word, special word,* $<$ *image* $>$ *token, term,* and *punctuation mark*. Standard words are unigram words in natural language. Special words [Sproat et al. 2001] include email address, IP address, URL, date, number, percentage, words containing special symbols (e.g., Ph.D., Prof.), unnecessary tokens (e.g., === and ###), etc. We identify special words using regular expressions. $<$ image $>$ tokens are $<$ image $>$ tags in the HTML file. We identify them by parsing the HTML file. Terms are base noun phrases extracted from the Web pages. We employed the methods proposed in Xun et al. [2000]. Punctuation marks include period, question, and exclamation marks.

(b) We assign tags to each token based on their corresponding type. For standard word, we assign all possible tags. For special words, we assign the following tags: *Position*, *Affiliation*, *Email*, *Address*, *Phone*, *Fax*, and *Bsdate*, *Msdate*, and *Phddate*. For the $<$ image $>$ token, we assign two tags: *Photo* and *Email* (it is likely that an email address is shown as an image). For term tokens, we assign *Position*, *Affiliation*, *Address*, *Bsmajor*, *Msmajor*, *Phdmajor*, *Bsuniv*, *Msuniv*, and *Phduniv*. In this way, each token can be assigned several possible tags. Using the tags, we can perform extraction of 16 profile properties, which cover 95.71% of the property values on the Web pages).

4.2 Extraction Model using Conditional Random Fields

We employ Conditional Random Fields (CRFs) as the tagging model. CRF is a special case of MRF. CRF is a conditional probability of a sequence of labels y given a sequence of observations tokens [Lafferty et al. 2001]. However, the previous linear-chain CRFs only model the linear dependencies as a sequence, but are not able to model hierarchical dependencies [Lafferty et al. 2001; Zhu et al. 2006].

In this section, we first introduce the basic concepts of Conditional Random Fields (CRFs) and the linear-chain CRFs, and then we explain a tree-structured CRF (TCRF) model to model the hierarchically laid-out information. Finally we discuss how to perform parameter estimation and extraction in TCRFs.

4.2.1 *Linear-Chain CRFs.* Conditional Random Fields are undirected graphical models [Lafferty et al. 2001]. As defined before, X is a random

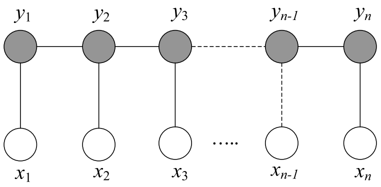


Fig. 4. Graphical representation of linear-chain CRFs.

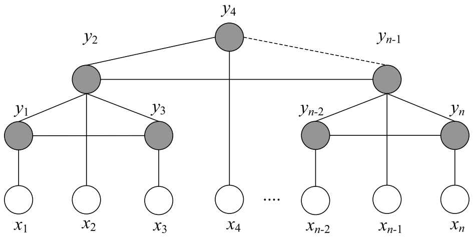


Fig. 5. Graphical representation of tree-structured CRFs.

variable over data sequences to be labeled, and Y is a random variable over corresponding label sequences. All components Y_i of Y are assumed to range over a finite label alphabet Y. CRFs construct a conditional model $P(Y|X)$ with a given set of features from paired observation and label sequences.

A CRF is a random field globally conditioned on the observation X. Linear-chain CRFs were first introduced by Lafferty et al. [2001]. An example of a graphical structure of linear-chain CRFs is shown in Figure 4.

By the fundamental theorem of random fields [Hammersley and Clifford 1971], the conditional distribution of the labels y given the observations data x has the form

$$P(y|x) = \frac{1}{Z(x)} exp(\sum_{e \in E, j} \lambda_j t_j(e, y|_e, x) + \sum_{v \in V, k} \mu_k s_k(v, y|_v, x)), \tag{1}$$

where x is a data sequence, y is a label sequence, and $y|_e$ and $y|_v$ are the set of components of y associated with edge e and vertex v in the linear chain, respectively; t_j and s_k are feature functions; parameters λ_j and μ_k correspond to the feature functions t_j and s_k, respectively, and are to be estimated from the training data; and $Z(x)$ is the normalization factor, also known as partition function.

4.2.2 *Tree-Structured Conditional Random Fields (TCRFs).* Linear-chain CRFs cannot model dependencies across hierarchically laid-out information. We propose a TCRF model [Tang et al. 2006]. The graphical structure of TCRFs is shown in Figure 5.

Table I. Definition of Information Block and Profile Properties

Block type	Profile property
Photo	Person photo
Basic information	Position, affiliation
Contact information	Fax, phone, address, email
Educational history	Phddate, Phduniv, Phdmajor, Msdate, Msuniv, Msmajor, Bsdate, Bsuniv, Bsmajor

From Figure 5, we see that y_4 is the parent vertex of y_2 and y_{n-1} (for a simplified description, hereafter we use parent-vertex to represent the upper-level vertex and child-vertex to represent the lower-level vertex). TCRFs can model the parent-child dependencies, for example, $y_4 - y_2$ and $y_4 - y_{n-1}$. Furthermore, y_2 and y_{n-1} are in the same level, which are represented as a sibling dependency in TCRFs.

Here we also use X to denote the random variable over observations, and Y to denote the corresponding labels. Y_i is a component of Y at the vertex i. In the same way as for linear-chain CRFs, we consider one vertex or two vertices as a clique in TCRFs. TCRFs can be also viewed as finite-state models. Each variable Y_i has a finite set of state values and we assume the one-to-one mapping between states and labels. Thus dependencies across components Y_i can be viewed as transitions between states.

$$P(y|x) = \frac{1}{Z(x)} exp(\sum_{c\in C,j} \lambda_j t_j(c, y|_c, x) + \sum_{v\in V,k} \mu_k s_k(v, y|_v, x)), \tag{2}$$

where c is a clique defined on edge (e.g., parent-child (y_p, y_c), child-parent (y_c, y_p), and sibling edge (y_s, y_s)) or triangle (e.g., (y_p, y_s, y_s)). t_j and s_k are feature functions.

TCRFs have the same form as linear-chain CRFs except that in TCRFs the edges include parent-child edges, child-parent edges, and sibling-vertices edges, while in linear-chain CRFs the edges mean the transitions from the previous state to the current state.

In researcher profile extraction, the observation x in TCRFs corresponds to the identified homepage/introducing page. The tree is obtained by converting the Web page into a DOM tree. The root node denotes the Web page, each leaf node in the tree denotes the word token, and the inner node denotes the coarse information block (e.g., a block containing contact information). The label y of the inner node thus corresponds one type of the coarse information block, while the label y of the leaf node corresponds to one of the profile properties. Definitions of the researcher profile properties and the coarse information block, as well their relationships, are summarized in Table I.

4.2.3 *Parameter Estimation.* The parameter estimation problem is to determine the parameters $\Theta = \{\lambda_1, \lambda_2, \cdots; \mu_k, \mu_{k+1}, \cdots\}$ from training data

$D = \{(x^{(i)}, y^{(i)})\}$ with empirical distribution. More specifically, we optimize the log-likelihood objective function with respect to a conditional model $P(y|x, \Theta)$:

$$L_{\Theta} = \sum_{i} \tilde{P}(x^{(i)}, y^{(i)}) \log P_{\Theta}(x^{(i)}, y^{(i)}). \tag{3}$$

In the following, to facilitate the description, we use f to denote both the edge feature function t and the vertex feature function s; use c to denote both edge e and vertex v; and use λ to denote the two kinds of parameters λ and μ. Thus, the derivative of the objective function with respect to a parameter λ_j associated with clique index c is

$$\frac{\partial L_{\Theta}}{\partial \lambda_j} = \sum_{i} \left[\sum_{c} f_j(c, y^{(i)}_{(c)}, x^{(i)}) - \sum_{y} \sum_{c} P(y_{(c)}|x^{(i)}) f_j(c, y^{(i)}_{(c)}, x^{(i)}) \right], \tag{4}$$

where $y^{i}_{(c)}$ is the label assignment to clique c in $x^{(i)}$, and $y_{(c)}$ ranges over label assignments to the clique c. We see that, for each clique, we need to compute the marginal probability $P(y_{(c)}|x^{(i)})$. The marginal probability $P(y_{(c)}|x^{(i)})$ can be again decomposed into $P(y_p, y_c|x^{(i)})$, $P(y_c, y_p|x^{(i)})$, $P(y_s, y_s|x^{(i)})$, and $P(y_i|x^{(i)})$, as we have three types of dependencies and one type of vertex. Moreover, we need to compute the global conditional probability $p(y^{(i)}|x^{(i)})$.

The marginal probabilities can be done using many inference algorithms for an undirected model (e.g., Belief Propagation [Yedidia et al. 2001]). However, as the graphical structure in TCRFs can be a tree with cycles, exact inference is infeasible. We propose using the Tree-based Reparameterization (TRP) algorithm [Wainwright et al. 2001] to compute the approximate probabilities of the factors. TRP is based on the fact that any exact algorithm for optimal inference on trees actually computes marginal distributions for pairs of neighboring vertices. For an undirected graphical model over variables x, this results in an alternative parameterization of the distribution as

$$P(x) = \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E} \psi_{st}(x_s, x_t) \Rightarrow P(x) = \prod_{s \in V} P_s(x_s) \prod_{(s,t) \in E} \frac{P_{st}(x_s, x_t)}{P_s(x_s) P_t(x_t)}, \tag{5}$$

where $\psi_s(x_s)$ is the potential function on single-vertex x_s, $\psi_{st}(x_s, x_t)$ is the potential function on edge (x_s, x_t), and Z is the normalization factor.

TRP consists of two main steps: initialization and updates. The updates are a sequence of $T_n \rightarrow T_{n+1}$ on the undirected graph with edge set E, where T represents the set of marginal probabilities maintained by TRP including single-vertex marginals $T^{n+1}_{u}(x_u)$ and pairwise joint distribution $T^{n+1}_{uv}(x_u, x_v)$; and n denotes the iteration number. The TRP algorithm is summarized in Figure 6.

So far, the termination conditions in the TRP algorithm are defined as follows: if the maximal change of the marginals is below a predefined threshold or the update times exceed a predefined number (defined as 1000 in our experiments), then stop the updates. When selecting spanning trees $R = \{\Gamma^i\}$, the only constraint is that the trees in R cover the edge set of the original undirected graph U. In practice, we select trees randomly, but we always first select edges that have never been used in any previous iteration.

(1) **Initialization**: for every node u and every pair of nodes (u, v), initialize T_u^0 by $T^0 = \kappa\psi_u$ and $T_{uv}^0 = \kappa\psi_{uv}$, with κ being a normalization factor.

(2) **TRP Updates**: for $i = 1, 2, \cdots$, do:

—Select some spanning tree Γ^i with edge set E^i, where $R = \{\Gamma^i\}$ is a set of spanning trees;

—Use any exact algorithm, such as belief propagation, to compute exact marginals $P^i(x)$ on Γ^i. For all $(u, v) \in E^i$, set

$$T_u^{i+1}(x_u) = P^i(x_u),\ T_{uv}^{i+1}(x_u, x_v) = \frac{P^i(x_u, x_v)}{P^i(x_u)P^i(x_v)};$$

—Set $T_{uv}^{i+1} = T_{uv}^i$ for all $(u, v) \in E \setminus E^i$ (i.e., all the edges not included in the spanning tree Γ^i);

—Stop if termination conditions are met.

Fig. 6. The TRP Algorithm.

Finally, to reduce overfitting, we define a spherical Gaussian weight prior $P(\Theta)$ over parameters, and penalize the log-likelihood object function as

$$L_\Theta = \sum_i P(x^{(i)}, y^{(i)}) \log P_\Theta(x^{(i)}, y^{(i)}) - \frac{\|\lambda\|^2}{2\sigma^2} + const \tag{6}$$

with gradient

$$\frac{\partial L_\Theta}{\partial \lambda_j} = \sum_i \left[\sum_c f_j(c, y_{(c)}^{(i)}, x^{(i)}) - logz(x^{(i)}) \right] - \frac{\lambda_j}{\sigma^2}, \tag{7}$$

where *const* is a constant.

The function L_Θ is convex, and can be optimized by any number of techniques, as in other maximum-entropy models [Lafferty et al. 2001]. In the result below, we used gradient-based L-BFGS [Liu et al. 1989], which has previously outperformed other optimization algorithms for linear-chain CRFs Sha and Pereira [2003].

4.2.4 *Extraction.* Extraction (also called *labeling*) is the task to find labels y^* that best describe the observations x, that is, $y^* = \max_y P(y|x)$. Dynamic programming algorithms, the most popular methods for this problem, can be used for extraction in TCRFs. We use the DOM tree presented in the Web page to infer the hierarchical structure. Then we use the TRP algorithm to compute the maximal value of $p(y|x)$.

4.2.5 *Features.* For each token unit, three types of features are defined: content features, pattern features, and term features.

4.2.5.1 *Content Features.* For a standard word, the content features include the following.

—*Word features.* Whether the current token is a standard word.

—*Morphological features.* The morphology of the current token, for example, whether the token is capitalized.

For a < image > token, the content features include the following.

—*Image size*. The size of the current image.

—*Image height/width ratio*. The ratio of the height to the width of the current image. The ratio of a person photo is likely to be greater than 1.0.

—*Image format*. The format of the image (e.g., JPG, BMP).

—*Image color*. The number of the "unique color" used in the image and the number of bits used for per pixel (e.g., 32, 24, 16, 8, and 1).

—*Face recognition*. Whether the current image contains a person face. We used a face recognition tool[3] to detect the person face.

—*Image filename*. Whether the image filename (partially) contains the researcher's name.

—*Image ALT*. Whether the alt attribute of the < image > tag (partially) contains the researcher's name.

—*Image positive keywords*. Whether the image filename contains positive keywords like myself.

—*Image negative keywords*. Whether the image filename contains negative keywords like logo.

4.2.5.2 *Pattern Features*. Pattern features are defined for each token.

—*Positive words*. Whether the current token contains positive *Fax/Phone* keywords like *Fax:*, *Phone:*, and positive *Position* keywords like *Manager*.

—*Special tokens*. Whether the current token is a special word.

4.2.5.3 *Term Features*. Term features are defined only for term token.

—*Term features*. Whether the token unit is a term.

—*Dictionary features*. Whether the term is included in a dictionary.

We can easily incorporate these features into our model by defining Boolean-valued feature functions. Finally, two sets of features are defined in the CRF model: transition features and state features. For example, a transition feature $y_{i-1} = y'$, $y_i = y$ implies that, if the current tag is y and the previous tag is y', then the value is true; otherwise it is false. The state feature $w_i = w$, $y_i = y$ implies that, if the token is w and the current tag is y, then the feature value is true; otherwise it is false. In total, 308,409 features were used in our experiments.

5. NAME DISAMBIGUATION

We crawled the publication data from existing online data sources. For integrating the researcher profiles and the publications data, we used researcher names and the author names as the identifier. The method inevitably has the name ambigity problem.

[3] `http://opencvlibrary.sf.net`

The goal of name disambiguation is to disambiguate n papers $P = \{p_1, p_2, \cdots, p_n\}$ that contain the author name a to k actual researchers $\{y_1, y_2, \cdots, y_k\}$ with respect to name a, that is, assigning an author label to each paper.

We propose a probabilistic model to deal with the problem. Our intuition in this method is based on two observations: (1) papers with similar content tend to have the same label (belonging to the same author); and (2) papers that have a strong relationship tend to have the same labels, for example, two papers are written by the same coauthors.

Our method is based on the Hidden Markov Random Field (HMRF) model, a special case of MRF. The reason we chose HMRF is due to its natural advantages. First, like all MRF family members, HMRF can be used to model dependencies (or relationships, e.g., CoAuthor) between observations (each paper is viewed as an observation). Second, HMRF supports unsupervised learning, supervised learning, and also semisupervised learning. In this paper, we will focus on unsupervised learning for name disambiguation using HMRF, but it is easy to incorporate some prior/supervised information into the model, thus extending the proposed approach to semisupervised learning. Third, it is natural to do model selection in the HMRF model. The objective function in the HMRF model is a posterior probability of hidden variables given observations, which can be used as a criterion for model selection.

In the rest of this section, we will introduce the hidden Markov Random Field model and then define the objective function for the name disambiguation problem.

5.1 Data Preparation

Each publication p_i has six attributes: paper title ($p_i.title$), publication venue ($p_i.pubvenue$), publication year ($p_i.year$), abstract ($p_i.abstract$), authors ($\{a_i^{(0)}, a_i^{(1)}, \cdots, a_i^{(u)}\}$), and references ($p_i.references$). We extracted the attribute values of each paper from several digital libraries, for example, from IEEE, Springer, and ACM. We used heuristics to perform the extraction.

We define five types of relationships between articles (Table II). Relationship r_1 represents two papers are published in the same venue. Relationship r_2 means two papers have a same secondary author, and relationship r_3 means one paper cites the other paper. Relationship r_4 indicates a constraint-based relationship supplied via user feedbacks. For instance, the user may specify that two papers should be disambiguated to the same author. We use an example to explain relationship r_5. Suppose p_i has authors David Mitchell and Andrew Mark, and p_j has authors David Mitchell and Fernando Mulford. We are going to disambiguate David Mitchell. If Andrew Mark and Fernando Mulford also coauthor another paper, then we say p_i and p_j have a *2-CoAuthor relationship*.

Specifically, to test whether two papers have a $\tau-$CoAuthor relationship, we construct a Boolean-valued matrix M, in which an element is 1 if its value is greater than zero; otherwise 0 (cf. Figure 7). In matrix M, $\{p_1, p_2, \cdots, p_n\}$ are publications with the principle author name a. $\{a_1, a_2, \cdots, a_p\}$ is the union set of all $p_i.authors \backslash a_i^{(0)}$, $i \in [1, n]$. Note that $\{a_1, a_2, \cdots, a_p\}$ does not include the

Table II. Relationships Between Articles

R	W	Relation name	Description
r_1	w_1	CoPubvenue	$p_i.pubvenue = p_j.pubvenue$
r_2	w_2	CoAuthor	$\exists r, s > 0, a_i^{(r)} = a_j^{(s)}$
r_3	w_3	Citation	p_i cites p_j or p_j cites p_i
r_4	w_4	Constraints	Feedbacks supplied by users
r_5	w_5	$\tau-$CoAuthor	$\tau-$extension coauthorship ($\tau > 1$)

$$\begin{array}{c|cccc|ccc}
 & p_1 & p_2 & \cdots & p_n & a_1 & \cdots & a_p \\
\hline
p_1 & 1 & 0 & \cdots & 0 & 1 & \cdots & 0 \\
p_2 & 0 & 1 & & & 0 & & 1 \\
\cdots & \cdots & & & & & & \\
p_n & 0 & & & 1 & 1 & & 0 \\
\hline
a_1 & 1 & 0 & & 1 & 1 & 0 & 1 \\
\cdots & \cdots & & & & & & \\
a_p & 0 & 1 & & 0 & 1 & 0 & 1
\end{array}
\qquad
\begin{array}{c|c|c}
 & p_1 \; p_2 \; \cdots \; p_n & a_1 \; \cdots \; a_p \\
\hline
\begin{array}{c} p_1 \\ p_2 \\ \cdots \\ p_n \end{array} & M_p & M_{pa} \\
\hline
\begin{array}{c} a_1 \\ \cdots \\ a_p \end{array} & M_{ap} & M_a
\end{array}$$

Fig. 7. Matrix M for r_5 relationship.

principle author name $a_i^{(0)}$. Submatrix M_p indicates the relationship between $\{p_1, p_2, \cdots, p_n\}$ and initially it is an identity matrix. In submatrix M_{pa}, an element on row i and column j is equal to 1 if and only if $a_j \in p_i.authors$; otherwise it is equal to zero. The matrix M_{ap} is symmetric to M_{pa}. Submatrix M_a indicates the coauthorship among $\{a_1, a_2, \cdots, a_p\}$. The value on row i and column j in M_a is equal to 1 if and only if a_i and a_j coauthor one paper in our database (not limited in $\{p_1, p_2, \cdots, p_n\}$) otherwise it is equal to zero. Then $\tau-$CoAuthor can be defined based on $M^{(\tau+1)}$, where $M^{(\tau+1)} = M^{(\tau)}M$ with $\tau > 0$.

The publication data with relationships can be modeled as a graph comprising of nodes and edges. Attributes of a paper are represented as a feature vector. In the vector, we use words (after stop words filtering and stemming) in the attributes of a paper as features and use occurring times as the values.

5.2 Formulation Using Hidden Markov Random Fields

Hidden Markov Random Field (HMRF) is a member of the family of MRF and its concept is derived from Hidden Markov Models (HMMs) [Ghahramani and Jordan 1997]. An HMRF is mainly composed of three components: an observable set of random variables $X = \{x_i\}_{i=1}^n$, a hidden field of random variables $Y = \{y_i\}_{i=1}^n$, and neighborhoods between each pair of variables in the hidden field.

We formalize the disambiguation problem as that of grouping relational papers into different clusters. Let the hidden variables Y be the cluster labels on the papers. Every hidden variable y_i takes a value from the set $\{1, \cdots, k\}$, which are the indexes of the clusters. The observation variables X correspond

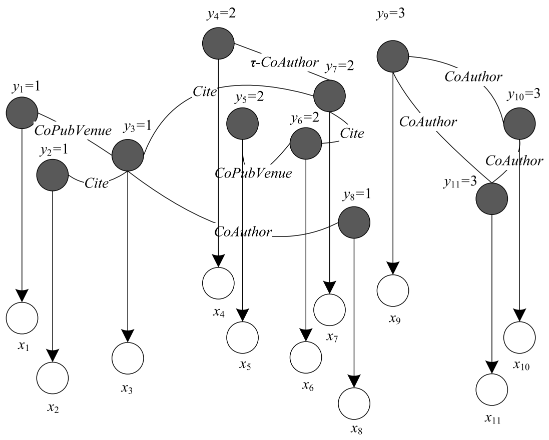


Fig. 8. Graphical representation of HMRF.

to papers, where every random variable x_i is generated from a conditional probability distribution $P(x_i|y_i)$ determined by the corresponding hidden variable y_i.

Figure 8 shows an example graphical representation of HMRF. The observation variable x_i corresponds to a paper and the hidden variable y_i corresponds to the assignment result. The dependent edge between the hidden variables corresponds to the relationship between papers (cf. Table II for the definition of the relationship).

By the fundamental theorem of random fields [Hammersley and Clifford 1971], the probability distribution of the label configuration Y has the form

$$P(Y) = \frac{1}{Z_1}\exp(\sum_{(y_i,y_j)\in E,k} \lambda_k f_k(y_i, y_j)) \tag{8}$$

and we assume the publication data is generated under the spherical Gaussian distribution; thus we have

$$P(X|Y) = \frac{1}{Z_2}\exp(\sum_{x_i\in X,l} \alpha_l f_l(y_i, x_i)), \tag{9}$$

where $f_k(y_i, y_j)$ is a nonnegative potential function (also called *feature function*) defined on edge (y_i, y_j) and E represents all edges in the graph; $f_l(y_i, x_i)$ is a potential function defined on node x_i; λ_k and α_l are weights of the edge feature function and the node potential (feature) function respectively; and Z_1 and Z_2 are normalization factors (also called *partition functions*).

We then discuss how to define the edge feature function and the node feature function. For the edge feature function $f_k(y_i, y_j)$, we define it by combining the different relationships between paper x_i and x_j:

$$f_k(y_i, y_j) = D(x_i, x_j) \sum_{r_m\in R_{ij}} w_m r_m(x_i, x_j). \tag{10}$$

Here, $D(x_i, x_j)$ is a similarity function between paper x_i and x_j; w_m is the weight of relationship r_m; R_{ij} denotes the set of relationships between x_i and x_j.

We define the node feature function as

$$f_l(y_i, x_i) = D(y_i, x_i) = D(\mu_{(i)}, x_i), \tag{11}$$

where $\mu_{(i)}$ is the representation of the researcher (also called *cluster centroid*) that the paper x_i is assigned to; it is defined as $\mu_h = \kappa \sum_{i, y_i = h} x_i$, with κ being a normalization factor. Notation $D(x_i, \mu_{(i)})$ represents the similarity between paper x_i and its assigned researcher $\mu_{(i)}$.

Now the problem is how to learn the model. In general, we can consider maximizing the following conditional log-likelihood $P(Y|X)$:

$$L_{\max} = \log(P(Y|X)) = \log(P(Y)P(X|Y)). \tag{12}$$

By substituting (8) and (9) into Equation (12), we obtain

$$L_{\max} = \log(P(Y|X)) = \log\left(\frac{1}{Z_1 Z_2}\exp\left[\sum_{(y_i,y_j)\in E,k} \lambda_k f_k(y_i, y_j) + \sum_{x_i \in X,l} \alpha_l f_l(y_i, x_i)\right]\right). \tag{13}$$

Then putting (10) and (11) into (13), we obtain

$$L_{\max} = \sum_{(y_i,y_j)\in E,k} \lambda_k D(y_i, y_j) r_k(x_i, x_j) + \sum_{x_i \in X,l} \alpha_l D(\mu_{(i)}, x_i) - \log Z, \tag{14}$$

where $Z = Z_1 Z_2$, and for simplicity of explanation, we use λ to denote the product of the weight λ_k of edge feature function and the weight w_m of the relationship.

5.3 Parameter Estimation

The parameter estimation problem is to determine the values of the parameters $\Theta = \{\lambda_1, \lambda_2, \cdots; \alpha_1, \alpha_2, \cdots\}$ and to determine assignments of all papers. More accurately, we optimize the log-likelihood objective function (14) with respect to a conditional model $P(Y|X, \Theta)$.

The algorithm for parameter estimation primarily consists of three steps: *initialization*, *assignment of papers*, and *update of parameters*. The basic idea is that we first choose an initialization of parameters Θ and select a centroid for each cluster. Then, we assign each paper to its closest cluster and then calculate the centroid of each cluster based on the assignments. After that, we update the weight of each feature function by maximizing the objective function. The iteration continues until convergence.

We define the similarity function $D(x_i, x_j)$ as follows (the function can be also defined in any other ways):

$$D(x_i, x_j) = 1 - \frac{x_i^T x_j}{\|x_i\| \|x_j\|}. \tag{15}$$

We now introduce in detail the three steps in the algorithm. In the *initialization* step, we first cluster publications into disjoint groups based on the relationships between them, that is, if two publications have a relationship, then they are assigned to the same researcher. Therefore, we get λ groups. If λ is equal to our actual researcher number k, then these groups are used as our initial assignment. If $\lambda < k$, we choose $k - \lambda$ random assignments. If $\lambda > k$, we cluster the nearest group until there are only k groups left.

In the *assignment* step, each paper x_i is assigned to $\mu_{(h)}$ to maximize the local $\log P(y_i|x_i)$:

$$\log P(y_i|x_i) = \sum_{(x_i,x_j)\in E_i,R_i,k} \lambda_k D(x_i, x_j) r_k(x_i, x_j) + \sum_l \alpha_l D(\mu_{(i)}, x_i) - \log(Z_x), \quad (16)$$

where E_i denotes all relationships related to x_i. The first two terms in Equation (16) are a polynomial combination of the local similarity function $D(x_i, \mu_{(h)})$ and the relational similarity function $D(x_i, x_j)$, which can be calculated in a polynomial time. Z_x is a normalization factor and can be approximately viewed as a constant.

The assignment of a paper is performed while keeping assignments of the other papers fixed. A greedy algorithm is used to sequentially update the assignment of each paper. The algorithm performs assignments in random order for all papers. The assignment process is repeated after all papers are assigned. This process runs until no paper changes its assignment between two successive iterations.

In the *update* step, each cluster centroid is first updated by the arithmetic mean of the papers contained in it, that is, $\mu_h = \kappa \sum_{i,y_i=h} x_i$, with κ being a normalization factor. Then the task is to update the parameters Θ. By differentiating the objective function with respect to each parameter λ_k, we have

$$\lambda_k^{new} = \lambda_k^{old} + \triangle \frac{\partial L}{\partial \lambda_k} = \lambda_k^{old} + \triangle \sum_{(x_i,x_j)\in E} D(x_i, x_j) r(x_i, x_j), \quad (17)$$

where $\triangle$ is the length of learning step.

6. USER INTEREST ANALYSIS

After extracting and integrating the user profiles, we obtain a basic user profile which consists of a set of profile properties and a set of documents for each user. Now we perform user interest analysis based on the user profile and its associated papers.

According to the definition of user interest in Section 2, our goal is to discover the latent topic distribution associated with each user. For different applications, the available information to discover the latent topic distribution is also different. As for the researcher profiling, our available information include publication venues, papers' contents, and authors.

Modeling the different information sources can be done in many different ways, for example, using the state-of-the-art language model (LM) [Baeza-Yates and Ribeiro-Neto 1999] or using a separated pLSI [Hofmann 1999] or LDA [Blei et al. 2003] for each type of object. However, the separated way may result in unsatisfactory performance. Some preliminary experimental results [Tang et al. 2008a] have confirmed confirm this assumption. Our main idea in

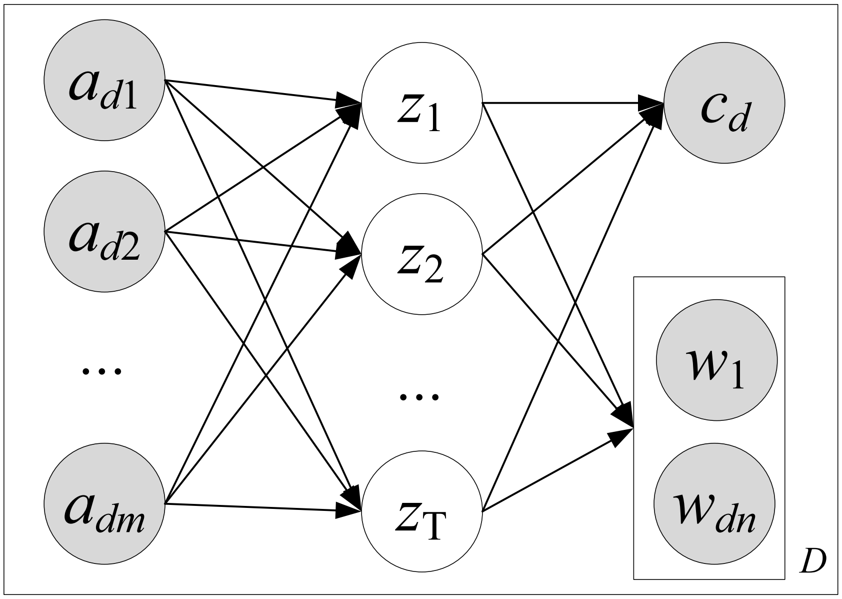


Fig. 9. Graphical representation of the Author-Conference-Topic (ACT) model.

this work is to use a probabilistic topic model to model papers, authors, and publication venues simultaneously.

The proposed topic model is called the *Author-Conference-Topic* (ACT) model. For simplicity, we use *conference* to denote all kinds of publication venues, including conferences, journals, and articles. Essentially, the model utilizes the topic distribution to represent the interdependencies among authors, papers, and publication venues. We considered three different strategies to implement the topic model in Tang et al. [2008b]. Here, we only introduce one implementation which achieves the best performance for academic search. The graphical representation of the ACT model is shown in Figure 9.

In the model, each author is associated with a multinomial distribution over topics and each word token in a paper and the conference stamp is generated from a sampled topic. The model actually reduces the process of writing a scientific paper to a series of probabilistic steps.

6.1 Notation

We define the notations used in this section. Assume that a paper d contains a vector $\mathbf{w}_d$ of N_d words, in which each word w_{di} is chosen from a vocabulary of size V, contains a vector $\mathbf{a}_d$ of A_d authors, with each author chosen from a set of authors of size A, and is published at the venue c_d, which is chosen from a set of publication venues of size C. Then a collection of D papers can be represented as $\mathbf{D} = \{(\mathbf{w}_1, \mathbf{a}_1, c_1), \cdots, (\mathbf{w}_D, \mathbf{a}_D, c_D)\}$. Table III summarizes the notations.

Table III. Notations

Symbol	Description
T	Number of topics
D	Number of papers
V	Number of unique words
A	Number of unique authors
C	Number of unique publication venues
N_d	Number of word tokens in paper d
A_d	Number of authors in paper d
$\mathbf{w}_d$	Vector form of word tokens in paper d
$\mathbf{a}_d$	Vector form of authors in paper d
c_d	The publication venue of paper d
w_{di}	The ith word token in paper d
z_{di}	The topic assigned to word token w_{di}
x_{di}	The chosen author associated with the word token w_{di}
θ_x	Multinomial distribution over topics specific to author x
ϕ_z	Multinomial distribution over words specific to topic z
ψ_z	Multinomial distribution over publication venues specific to topic z
α, β, μ	Dirichlet priors to multinomial distribution θ, ϕ, and ψ, respectively

6.2 Proposed Model

In the ACT model, the conference information is associated with each word as a stamp. For generating a word w_{di} in paper d, an author x_{di} is first chosen uniformly to be responsible for the word. Each author is associated with a *topic distribution*. Then a topic is sampled from the author-specific topic distribution. Next, the word and the conference stamp is sampled from the chosen topic. The intuition behind this model can be explained as follows: when preparing a paper, an author would be responsible for a word; he writes the word based on his research interests (i.e., *the associated topic distribution*); then each topic in this paper determines a proportion on where to publish the paper (i.e., *sampling the conference stamp from the topic*). Formally, the generative process can be described as follows.

(1) For each topic z, draw ϕ_z and ψ_z respectively from Dirichlet prior β and μ;

(2) For each word w_{di} in paper d:
—draw an author x_{di} from $\mathbf{a}_d$ uniformly;
—draw a topic z_{di} from a multinomial distribution $\theta_{x_{di}}$ specific to author x_{di}, where θ is generated from a Dirichlet prior α;
—draw a word w_{di} from multinomial $\phi_{z_{di}}$;
—draw a conference stamp c_{di} from multinomial $\psi_{z_{di}}$.

For inference, the task is to estimate the unknown parameters in the topic model. There are two sets of unknown parameters: (1) the distribution θ of A author topics, the distribution ϕ of T topic words, and the distribution ψ of T topic-conferences; and (2) the corresponding topic z_{di} and author x_{di} for each word w_{di}. It is usually intractable to compute the exact inference in such probabilistic model. A variety of algorithms have been proposed to conduct approximate inference, for example, variational EM methods [Blei et al. 2003], Gibbs sampling [Griffiths and Steyvers 2004; Steyvers et al. 2004], and expectation propagation [Griffiths and Steyvers 2004; Minka 2003]. We choose Gibbs sampling for its ease of implementation. Additionally, instead of estimating the model parameters directly, we evaluate the posterior distribution on just x and z and then use the results to infer θ, ϕ, and ψ. Specifically, we begin with the joint probability of the entire data set:

$$P(\mathbf{x}, \mathbf{z}, \mathbf{w}, \mathbf{c}|\Theta, \Phi, \Psi, \mathbf{a}) = \prod_{d=1}^{D} \prod_{i=1}^{N_d} \frac{1}{A_d} \times \prod_{x=1}^{A} \prod_{z=1}^{T} \left(\theta_{xz}^{m_{xz}} \prod_{v=1}^{V} \phi_{zv}^{n_{zv}} \prod_{c=1}^{C} \psi_{zc}^{n_{zc}} \right), \tag{18}$$

where m_{xz} is the number of times that topic z has been associated with the author x; n_{zv} is the number of times that word w_v was generated by topic z; and n_{zc} is the number of times that conference c was generated by topic z.

By placing a Dirichlet prior over Θ and another two over Φ and Ψ, and combining them into Equation (18) with further integrating over Θ, Φ, and Ψ, we obtain the probability $P(\mathbf{x}, \mathbf{z}, \mathbf{w}, \mathbf{c}|\alpha, \beta, \mu, \mathbf{a})$. Then by using the chain rule, we obtain the posterior probability of sampling the topic z_{di} and the author x_{di} for the word w_{di}:

$$\begin{aligned} &P(z_{di}, x_{di}|\mathbf{z}_{-di}, \mathbf{x}_{-di}, \mathbf{w}, \mathbf{c}, \alpha, \beta, \mu) \\ &\propto \frac{m^{-di}_{x_{di}z_{di}} + \alpha_{z_{di}}}{\sum_z (m^{-di}_{x_{di}z} + \alpha_z)} \frac{n^{-di}_{z_{di}w_{di}} + \beta_{w_{di}}}{\sum_v (n^{-di}_{z_{di}v} + \beta_v)} \frac{n^{-d}_{z_{di}c_d} + \mu_{c_d}}{\sum_c (n^{-d}_{z_{di}c} + \mu_c)}, \end{aligned} \tag{19}$$

where $\mathbf{z}_{-di}$ and $\mathbf{x}_{-di}$ represent all topics and authors assignments excluding the ith word in the paper d; and the number m^{-di} and n^{-di} with the superscript $-di$ denote a quantity, excluding the current instance (the ith word token or the conference stamp in the paper d).

As for the hyperparameters α, β, and μ, one could estimate the optimal values by using a Gibbs EM algorithm [Andrieu et al. 2003; Minka 2003]. For some applications, topic models are sensitive to the hyperparameters and it is necessary to get the right values. In the applications discussed in this work, we have found that the estimated topic models are not very sensitive to the hyperparameters. Thus, for simplicity, we took a fixed value (i.e., $\alpha = 50/T$, $\beta = 0.01$, and $\mu = 0.1$).

6.3 User Interest Discovery

After estimating the topic model, we can obtain a multinomial distribution $\theta_x = \{P(z|x)\}_z$ over topics specific to each author x. Further each topic is represented by a multinomial distribution over words and another multinomial distribution over conferences. The distribution θ_x, the mixture of topics, is taken as the user interest.

Formally, each topic z is represented as a set of word-probability pairs $\{(w_i, p(w_i|z))\}_i$ or $\{\phi_{zw_i}\}_i$ and a set of conference-probability pairs $\{(c_k, p(c_k|z))\}_k$ or $\{\psi_{zc_k}\}_k$. Finally the user a's interest is represented as a set of topic-probability pairs $\{(z, p(z|a))\}_z$ or $\{\theta_{az}\}_z$.

6.4 Parallelization

As the Gibbs sampling algorithm for parameter estimation needs to make multiple passes over the entire dataset, it often takes multiple days (even weeks) to learn the topic model on a large-scale dataset, which makes it not practical for many applications.

Inspired by the distributed inference for LDA [Newman et al. 2007], we can implement a distributed inference algorithm over multiple processors for the proposed models. The basic idea is to conduct the inference in a "distribute-and-merge" way. In distribution, given P processors, we distribute the document collection D over the P processors, with $D_p = D/P$ documents on each processor. Then we partition the author-specific (author by topic) count matrix to the P processors and duplicate the other (topic by word, topic by conference) matrices to each processor. For parameter estimation, we conduct Gibbs sampling on each processor for the distributed documents for a number of internal iterations independently. In the internal iteration, the duplicated matrices will be updated independently. In merge, we combine the count matrices to guarantee the consistence of the count matrixes. More accurately, we respectively update each element of three duplicated (topic by word, topic by conference) matrices by

$$n_{zv}^{(new)} = n_{zv}^{(old)} + \sum_{p=1}^{P}(n_{zv}^{(p)} - n_{zv}^{(old)}), \tag{20}$$

$$n_{zc}^{(new)} = n_{zc}^{(old)} + \sum_{p=1}^{P}(n_{zc}^{(p)} - n_{zc}^{(old)}), \tag{21}$$

where the number $n^{(old)}$ with the superscript (old) denotes the count before distribution and the number $n^{(new)}$ with the superscript (new) denotes the count after merge. The number $n^{(p)}$ denotes the count obtained after the independent sampling on each processor. The distributed inference algorithm can be considered as an approximation of the single-processor inference algorithm.

6.5 Computational Complexity

We estimate the ACT model in an offline mode. The model has a complexity of $O(MD\bar{N}_dT\bar{A}_d)$, where M is the number of sampling times, $\bar{N}_d$ is the average number of word tokens in a paper, and $\bar{A}_d$ is the average number of authors. In most cases, the number $\bar{A}_d$ is negligible to the final complexity. In the parallelized ACT model, the time complexity is $O(M((D/P)D\bar{N}_dT\bar{A}_d) + (M/I_p)(TV + TC))$, where I_p is number of internal iterations on each processor; $(M/I_p)(TV + TC)$ is the time complexity of duplicating and merging the matrices to/from each processor. We see that, with the parallelization over multiple processors (e.g., 100 processor) and with an appropriate number of the internal iteration (e.g., 10), we can obtain a significant reduction of the time complexity.

7. EXPERIMENTAL RESULTS

We evaluated the proposed models in the context of the ArnetMiner system.[4] We conducted four experiments to evaluate the proposed approaches: profile

[4] http://arnetminer.org

extraction, name disambiguation, user interest analysis, application to expert finding.

7.1 Profile Extraction Performance

7.1.1 *Data Sets and Baselines.* For evaluating our profiling method, we randomly chose in total 1000 researcher names from the ArnetMiner database. We used the method described in Section 4 to find the researchers' homepages or introducing pages. If the method could not find a Web page for a researcher, we removed the researcher name from the data set. We finally obtained 898 Web pages. Seven human annotators conducted annotation on the Web pages. A spec was created to guide the annotation process. For disagreements in the annotation, we conducted "majority voting". The spec and the dataset are publicly available.[5]

In the experiments, we conducted evaluations in terms of precision, recall, and $F1$ measure (for definitions of the measures, see, e.g., van Rijsbergen [1979]).

We defined baselines for researcher profile extraction. We used the rule learning, the classification based approach, and the linear-chain CRF as baselines. For the former approach, we employed the Amilcare system [Ciravegna 2001]. The system is based on a rule induction algorithm, called LP^2. For the classification-based approach, we trained a classifier to identify the value of each property. We employed support vector machines (SVMA) [Cortes and Vapnik 1995] as the classification model. For the linear-chain CRF, we implemented the algorithm described in Lafferty et al. [2001], which is also publicly available.[6]

7.1.2 *Results.* Table IV shows the five-fold cross-validation results. Our method clearly outperforms the baseline methods (+29.93% better than Amilcare, +9.80% better than SVM, and +3.33% better than CRF, respectively, in terms of $F1$ score).

We conducted sign tests for each extraction subtask, which showed that all the improvements of the proposed TCRF over Amilcare, SVM, and CRF are statistically significant ($p \ll 0.01$).

7.1.3 *Discussion.* Our method outperforms the baseline methods on almost all profile properties, especially those having strong dependencies with each other.

The baseline methods suffer from ignorance of dependencies between the subtasks. For example, 1460 cases (19.31%) of *Affiliation* need to use the results of *Position*, and 17.66% of the educational history information (e.g., *Phdmajor* and *Phduniv*) need use dependencies with each other. However, the baseline methods cannot make use of the dependencies, as it conducts all the subtasks from raw input data.

[5] http://arnetminer.org/lab-datasets/profiling/index.html

[6] http://keg.cs.tsinghua.edu.cn/persons/tj/software/KEG_CRF/

Table IV. Performances of Researcher Profile Extraction (%)

Profiling Task	TCRF			CRF			SVM			Amilcare		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Photo	94.71	88.46	91.48	90.32	88.09	89.11	87.99	89.98	88.86	97.44	52.05	67.86
Position	85.15	71.27	77.59	77.53	63.01	69.44	78.62	55.12	64.68	37.50	61.71	46.65
Affiliation	85.81	88.91	87.33	84.21	82.97	83.52	78.24	70.04	73.86	42.68	81.38	55.99
Phone	90.86	94.62	92.70	89.78	92.58	91.10	77.91	81.67	79.71	55.79	72.63	63.11
Fax	92.65	89.74	91.17	92.51	89.35	90.83	77.18	54.99	64.17	84.62	79.28	81.86
Email	90.2	83.83	86.90	81.21	82.22	80.35	93.14	69.18	79.37	51.82	72.32	60.38
Address	92.35	86.27	89.21	87.94	84.86	86.34	86.29	69.62	77.04	55.68	76.96	64.62
Bsuniv	62.9	77.23	69.33	74.44	62.94	67.38	86.06	46.26	59.54	21.43	20.00	20.69
Bsmajor	65.91	54.72	59.80	73.20	58.83	64.20	85.57	47.99	60.75	53.85	18.42	27.45
Bsdate	70.83	40.48	51.52	62.26	47.31	53.49	68.64	18.23	28.49	17.95	16.67	17.28
Msuniv	64.88	52.87	58.26	66.51	51.78	57.55	89.38	34.77	49.78	15.00	8.82	11.11
Msmajor	78.57	70.97	74.58	69.29	59.03	63.35	86.47	49.21	62.10	45.45	20.00	27.78
Msdate	69.23	56.25	62.07	57.88	43.13	48.96	68.99	19.45	30.07	30.77	25.00	27.59
Phduniv	81.34	57.67	67.49	71.22	58.27	63.73	82.41	43.82	57.01	23.40	14.29	17.74
Phdmajor	78.87	74.67	76.71	77.55	62.47	67.92	91.97	44.29	59.67	68.57	42.11	52.17
Phddate	77.5	54.39	63.92	67.92	51.17	57.75	73.65	29.06	41.44	39.13	15.79	22.50
Overall	88.37	85.1	86.70	84.98	81.90	83.37	81.66	66.97	73.57	48.60	59.36	53.44

Here we use an example to show the advantage of our method compared with the methods without utilizing dependencies. The input text is (< tag > and < /tag > are labeled tags) as follows:

He received a B.A. in < bsmajor >Philosophy< /bsmajor > from < bsuniv >Oberlin College< /bsuniv > in < bsdate >1984< /bsdate >, and a Ph.D. from the Department of < phdmajor >Computer Science< /phdmajor > at the < phduniv >University of Maryland< /phduniv > in < phddate >1992< /phddate >.

With extraction by Amilcare, we obtain

He received a B.A. in < bsmajor >Philosophy< /bsmajor > from < bsuniv >Oberlin College< /bsuniv > in < bsdate >1984< /bsdate >, and a Ph.D. from the Department of < phdmajor >Computer Science< /phdmajor > at the **University of Maryland** in < phddate >1992< /phddate >.

With extraction by SVM, we obtain

He received a B.A. in < bsmajor >Philosophy< /bsmajor > from < bsuniv >Oberlin College< /bsuniv > in < **phddate** >**1984**< **/phddate** >, and a Ph.D. from the Department of < phdmajor >Computer Science< /phdmajor > at the < phduniv >University of Maryland< /phduniv > in **1992**.

With extraction by the proposed approach, we obtain

He received a B.A. in < bsmajor >Philosophy< /bsmajor > from < bsuniv >Oberlin College< /bsuniv > in < bsdate >1984< /bsdate >, and a Ph.D. from the Department of < phdmajor >Computer Science< /phdmajor > at the < phduniv >University of Maryland< /phduniv > in < phddate >1992< /phddate >.

Amilcare can extract some properties correctly. However, it does not recognize University of Maryland as *Phduniv*. SVM can extract some of the properties correctly, as well. For instance, it can detect the *Bsmajor* and *Bsuniv*. However, it mistakenly identifies 1984 as the *Phddate* and cannot identify the *Phddate* 1992. The proposed TCRF can take advantage of the hierarchical dependencies among the profile properties and thus correct 3.78% of the errors that linear-chain CRF cannot handle. Similarly, our method can correct 10.18% of the errors that SVM cannot handle and 22.13% of the errors that Amilcare cannot handle.

We also conducted error analysis on the results of our method. Table V is a confusion matrix of the results. We found that a large number of errors are cases that profile properties are recognized as *Other*. This problem possibly can be solved by incorporating more useful features in the proposed approach. However, manually defining the features would be tedious. A potential way is to employ an automatic method to learn some (non-)consecutive string patterns as the features. We tried the method proposed in Cao et al. [2003]. However, the current result is not satisfactory. Another type of error is introduced by the confusion among similar profile properties, for example *Position*, *Affiliation*, and *Address*. This implies that the linear and tree-structured dependencies used in the proposed approach are still not sufficient. Further analysis shows that some properties can be only disambiguated by using long-distance dependencies (e.g., dependencies between sections). How to further

Table V. Confusion Matrix on All Subtasks of Profile Extraction

	Other	Pic.	Pos.	Aff.	Add.	Bsm	Bsu	Bsd	Msm	Msu	Msd	Phdm	Phdu	Phdd	Em	Ph.	Fax
Pic.	119	778	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Pos.	560	0	1023	140	11	0	0	0	0	0	0	5	0	0	0	0	0
Aff.	749	0	33	5969	424	7	8	0	2	6	0	21	37	0	0	0	0
Add.	612	0	0	429	6266	0	0	0	0	0	0	0	0	0	10	0	0
Bsm	102	0	0	22	0	144	0	0	16	0	0	15	0	0	0	0	0
Bsu	106	0	0	23	0	6	159	0	0	21	0	0	13	0	0	0	0
Bsd	136	0	0	0	0	0	0	60	0	0	10	0	0	11	0	0	0
Msm	123	0	0	7	0	6	0	0	90	8	0	13	0	0	0	0	0
Msu	96	0	0	16	0	5	17	0	3	114	0	0	23	0	0	0	0
Msd	140	0	0	0	0	0	0	22	0	0	30	0	0	8	0	0	0
Phdm	141	0	4	18	0	1	0	0	9	0	0	264	1	0	0	0	0
Phdu	133	0	0	56	0	0	5	0	0	6	0	7	255	0	0	0	0
Phdd	158	0	0	0	0	0	0	7	0	0	4	0	0	99	0	0	0
Em	392	3	0	0	20	0	0	0	0	0	0	0	0	0	1610	0	0
Ph.	136	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4301	158
Fax	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	302	2462

Table VI. Abbreviate Name Dataset

Abbr. name	# Publications	#Actual persons	Abbr. name	# Publications	#Actual persons
B. Liang	55	13	M. Hong	69	17
H. Xu	189	59	W. Yang	249	78
K. Zhang	320	40			

Table VII. Real Name Dataset

Person name	# Publications	#Actual persons	Person name	# Publications	#Actual persons
Cheng Chang	12	3	Gang Wu	40	16
Wen Gao	286	4	Jing Zhang	54	25
Yi Li	42	21	Kuo Zhang	6	2
Jie Tang	21	2	Hui Fang	15	3
Bin Yu	66	12	Lei Wang	109	40
Rakesh Kumar	61	5	Michael Wagner	44	12
Bing Liu	130	11	Jim Smith	33	5

incorporate long-distance dependencies into the proposed approach is also part of our ongoing work.

7.2 Name Disambiguation Performance

7.2.1 *Data Sets and Evaluation Measure.* To evaluate our methods, we created two data sets from ArnetMiner, namely Abbreviate Name dataset and Real Name dataset. The first data set was collected by querying five abbreviated names in our database. All these abbreviated names are generated by simplifying the original names to its first name initial and last name. For example, Cheng Chang is simplified to C. Chang. Statistics of this data set are shown in Table VI.

Another data set includes 14 real person names. In these names, some names only correspond to a few persons. For example Cheng Chang corresponds to three actual persons and Wen Gao to four. However, some names seem to be popular. For example, there are 25 persons with the name Jing Zhang and 40 persons with Lei Wang. Statistics of this data set are shown in Table VII.

Five human annotators conducted disambiguation on the papers. A spec was created to guide the annotation process. Each paper was labeled with a number indicating the actual person. The labeling work was carried out based on the publication lists, affiliations, and email addresses on the authors' homepages. For further disagreements in the annotation, we conducted "majority voting."

From the statistics, we found that the disambiguation results were very unbalanced. For example, there were 286 papers authored by Wen Gao, with 282 of them authored by Prof. Wen Gao from the Institute of Computing at the Chinese Academy of Science and only four papers authored by the other three Wen Gaos.

We defined a baseline method based on the hierarchical clustering algorithm. The method is similar to that proposed by Tan et al. [2006] except that Tan et al. [2006] also utilized a search engine to help the disambiguation. We also compared our approach with the method proposed in Yin et al.

Table VIII. Results on Name Disambiguation (%)

Dataset	Person name	Baseline Tan et al. [2006]			Our approach		
		Prec.	Rec.	F1	Prec.	Rec.	F1
Abbr. name	B. Liang	82.07	76.90	79.07	49.54	100.00	66.26
	H. Xu	65.87	59.48	71.27	32.77	100.00	49.37
	K. Zhang	75.67	60.27	67.84	71.03	100.00	83.06
	M. Hong	79.24	65.36	71.36	91.32	86.06	88.61
	W. Yang	71.30	62.83	66.99	52.48	99.86	68.81
Avg.		74.43	64.47	69.21	59.43	97.18	73.75
Real name	Cheng Chang	100.00	100.00	100.00	100.0	100.0	100.0
	Wen Gao	96.60	62.64	76.00	99.29	98.59	98.94
	Yi Li	86.64	95.12	90.68	70.91	97.50	82.11
	Jie Tang	100.0	100.0	100.0	100.0	100.0	100.0
	Gang Wu	97.54	97.54	97.54	71.86	98.36	83.05
	Jing Zhang	85.00	69.86	76.69	83.91	100.0	91.25
	Kuo Zhang	100.0	100.0	100.0	100.0	100.0	100.0
	Hui Fang	100.0	100.0	100.0	100.0	100.0	100.0
	Bin Yu	67.22	50.25	57.51	86.53	53.00	65.74
	Lei Wang	68.45	41.12	51.38	88.64	89.06	88.85
	Rakesh Kumar	63.36	92.41	75.18	99.14	96.91	98.01
	Michael Wagner	18.35	60.26	28.13	85.19	76.16	80.42
	Bing Liu	84.88	43.16	57.22	88.25	86.49	87.36
	Jim Smith	92.43	86.80	89.53	95.81	93.56	94.67
Avg.		82.89	78.51	80.64	90.68	92.12	91.39

[2007]. In all experiments, we suppose that the number of persons k was provided manually.

7.2.2 *Results.* We evaluated the performances of our method and the baseline methods on the two datasets. Table VIII shows the results. It can be seen that our method outperforms the baseline method for name disambiguation (+4.54% better on the Abbr. Name dataset and +10.75% better on the Real Name dataset in terms of the average $F1$ score).

The baseline method suffers from two disadvantages: (1) it cannot take advantage of relationships between papers and (2) it relies on a fixed-distance measure. Our framework benefits from the ability of modeling dependencies between assignment results.

We compared our approach with the approach DISTINCT proposed in Yin et al. [2007]. We used the person names that were used in both Yin et al. [2007] and our experiments for comparison. Figure 10 shows the comparison. It can be seen that for some names (e.g., Hui Fang and Rakesh Kumar), both DISTINCT and the proposed approach achieved high performance. This is because papers of these names were clearly separated. (Cf. Section 7.2.4 for a more detailed distribution analysis.) While with some names our approach outperformed DISTINCT (e.g., Michael Wagner), with some other names our approach underperformed DISTINCT (e.g., Bin Yu). This is because DISTINCT determines the weight of each link (e.g., coauthor relationship) by supervised learning from an automatically constructed training set. The weights correspond to λ in Equation (14). The difference is that our approach learns the weights in a unsupervised way. The learned weights have different preferences on datasets

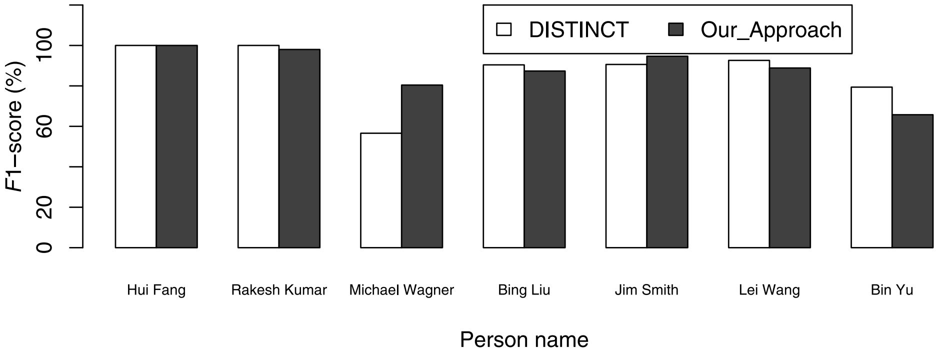


Fig. 10. Comparison with existing method.

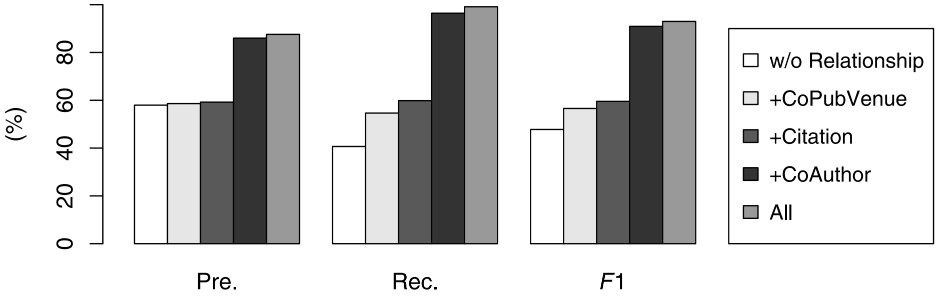


Fig. 11. Contribution of relationships.

with different distributions. We manually tuned the weights and found that the performance varied correspondingly. The current configuration of the parameters in our approach can result in a near best performance on average.

7.2.3 *Contribution of Relationships.* We further analyzed the contribution of different relationships for name disambiguation. We first evaluated the results of our approach by removing all relationships. Then we added the following relationships: CoConference, Citation, CoAuthor, and $\tau-$CoAuthor into our approach one by one. Figure 11 shows the results. *w/o Relationships* denotes our approach without any relationships. *+CoConference* denotes the results of by adding CoConference relationships. Likewise for the others. At each step, we observed improvements in terms of the $F1$ score. We should note that without using relationships the performances drop sharply (−15.65% on Abbr. Name and −44.72% on Real Name). This confirms that a framework for integrating relationships for name disambiguation is needed and each defined relationship in our method is helpful.

We can also see that the CoAuthor relationship makes major contributions (+24.38% by $F1$) to the improvements. CoPubvenue and Citation make limited contributions (+0.68% and +0.61%, respectively), to the improvements on

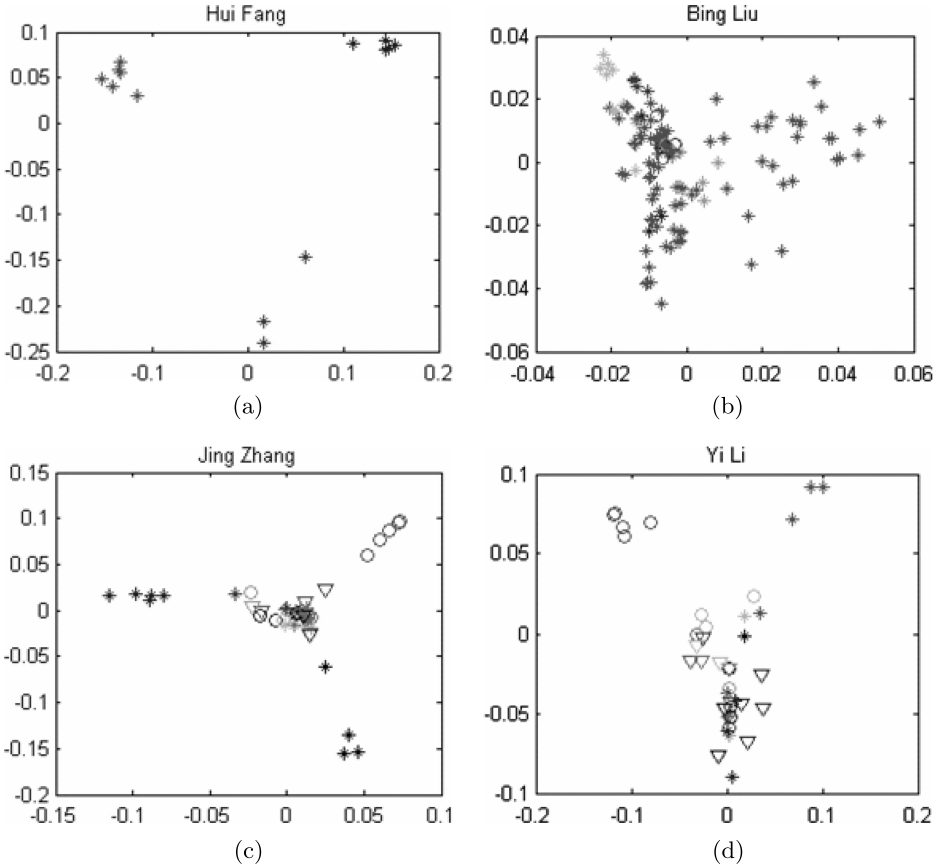


Fig. 12. Distribution analysis.

precision, but can obtain improvements (+13.99% and +5.20%, respectively) on recall.

7.2.4 *Distribution Analysis.* Figure 12 shows several typical feature distributions in our data sets. The graphs were generated using a dimension reduction method described in Cai et al. [2007]. The distributions can be typically categorized into (1) papers of different persons are clearly separated (Hui Fang, in Figure 12(a)). Name disambiguation on this kind of data can be solved pretty well by our approach and as well as by the baseline method; (2) publications are mixed together; however, there is a dominant author who writes most of the paper (e.g., Bing Liu, in Figure 12(b)); our approach can achieve an $F1$ score of 87.36%; however, the baseline method results in low accuracy (57.22% by $F1$); and (3) publications of different authors are mixed (Jing Zhang and Yi Li, in Figure 12(c) and 12(d)). Our method can obtain 92.15% and 82.11% in terms of the $F1$ measure; while the baseline method can only obtain 76.69% and 90.68% in terms of the $F1$ measure, respectively.

Table IX. Research Interest Analysis: Research Topics, Top 10 Representative Words, and Top 10 Conferences Found by ACT

Raymond J. Mooney		W. Bruce Croft	
Topic 162: Machine learning	0.6413	Topic 75: Information retrieval	0.8943
Topic 180: Multiagent reasoning	0.2356	Topic 24: Database systems	0.0387
Topic 199: Statistical machine translation	0.0801		
Learning	0.053442	Information	0.020554
Information	0.029767	Web	0.017087
Extraction	0.022361	Learning	0.016322
Web	0.014841	Text	0.014615
Semantic	0.009696	Classification	0.014315
Data	0.008753	Retrieval	0.011971
Text	0.008360	Search	0.009458
Approach	0.007947	Approach	0.008860
Logic	0.007518	Model	0.007995
Rules	0.007229	Data	0.007953
AAAI	0.190748	SIGIR	0.104724
IJCAI	0.126281	CIKM	0.099845
Machine Learning	0.053669	Inf. Process. Manage.	0.024329
ICML	0.049556	AAAI	0.023232
KDD	0.038491	ECIR	0.022895
JAIR (JAIR)	0.029873	ICML	0.017832
ACL	0.028894	JASIST	0.016286
ECML	0.024173	IJCAI	0.013898
Artif. Intell.	0.022655	ACM Trans. Inf. Syst.	0.012742
IIWeb	0.017081	ACL	0.011970

7.3 User Interest Analysis

We performed topic model estimation on the entire ArnetMiner data (448,365 researcher and 2,725,343 papers). We empirically set the number of topics as $T = 200$ for the ACT model. One can also use some solution like Teh et al. [2004] used to automatically estimate the number of topics. We ran five independent Gibbs sampling chains for 2,000 iterations each. A complete topic modeling result can be found online.[7]

Table IX shows example interests of two researchers (Raymond J. Mooney and W. Bruce Croft) discovered by the topic modeling method. Each author is associated with multiple research topics, 10 representative words, and the top 10 conferences with their corresponding probabilities. The results can be directly used to characterize the researcher interests more accurately than using keywords only in the traditional methods. They can also be used for prediction/suggestion tasks. For example, one can use the modeling results to find the best matching reviewer for a specific conference. Previously, such work was fulfilled by keyword matching only or topic-based retrieval such as in Mimno and McCallum [2007], but not considering the conference information. However, in practice, conference information is very important in finding the most appropriate reviewers.

[7] http://arnetminer.org/topicBrowser.do

Here, we discuss about the interdependencies between the subtasks of profiling. In general, the extracted profiles can be used to help name disambiguation. Assuming that there exists an accurately extracted profile base, we can use the number of profiles with a specific person name as the initial value for the person number *k* for name disambiguation. We can also use the affiliation and position information to help improve the accuracy of disambiguation. On the contrary, the disambiguation results can be also used to help identify the homepage of a researcher. Additionally, the results of profile extraction and name disambiguation would be very helpful for research interest analysis. For example, with the disambiguated papers for a person, we can estimate a person-specific interest distribution, instead of the name-specific one used in most traditional methods. The three subtasks are intertwined. An ideal solution for user profiling might be a model that can solve all the tasks together. However, the different tasks involve many different factors at different levels and a unified model may contain many complex variables, which makes the model prone to overfitting.

7.4 Expert Finding Experiments

To further evaluate the effectiveness of our method, we applied it to expert finding. The task of expert finding is to identify persons with some expertise or experience on a specific topic (query) in a social network. One common practice of finding information using social networks is to ask an expert, and thus expert finding plays an important role in social networks. We designed this experiment to see whether the annotated profile information can improve the performance of expert finding.

We collected 44 queries from the query log of ArnetMiner for evaluation purposes. Specifically, we selected the most frequent queries from the log of ArnetMiner (by removing overly specific or lengthy queries, e.g., “A Convergent Solution to Tensor Subspace Learning”). We also normalized similar queries (e.g., “Web Service” and “Web Services” to “Web Service”).

We conducted the experiments on a subset of the data (including 14, 134 persons, 10, 716 papers, and 1, 434 conferences) from ArnetMiner. For evaluation, it is difficult to find a standard data set with ground truth. As a result, we used the method of pooled relevance judgments [Buckley and Voorhees 2004] together with human judgments. Specifically, for each query, we first pooled the top 30 results from three similar (academic search) systems (Libra, Rexa, and ArnetMiner) into a single list. Then two faculties and five graduate students from CS provided human judgments. Four grade scores (3, 2, 1, and 0) were assigned, respectively representing definite expertise, expertise, marginal expertise, and no expertise. For example, for annotating persons, assessments were carried out mainly in terms of how many publications she/he had published related to the given query, how many top conference papers he or she had published, and what distinguished awards she/he had been awarded. Finally, the judgment scores were averaged to construct the final truth ground. The dataset was previously used in Tang et al. [2008a,b] and Zhang et al. [2007b] and is also online available.

In all the experiments, we conducted evaluation in terms of P@5, P@10, P@20, R-pre, and mean average precision (MAP). Readers are referred to Buckley and Voorhees [2004] and Craswell et al. [2005] for details of the measures.

We use language model (LM) [Zhai and Lafferty 2001] as the baseline method. We combined all documents authored by a researcher as a virtual document d. Then we used the following equations to calculate the relevance of each virtual document with the query q:

$$P(q|d) \ = \ \Pi_{w \in q} P(w|d), \tag{22}$$

$$P(w|d) \ = \ \frac{N_d}{N_d + \lambda} \cdot \frac{tf(w,d)}{N_d} + \left(1 - \frac{N_d}{N_d + \lambda}\right) \cdot \frac{tf(w,\mathbf{D})}{N_{\mathbf{D}}}, \tag{23}$$

where N_d is the number of word tokens in the virtual document d, $tf(w,d)$ is the frequency of word w in d, $|N_{\mathbf{D}}$ is the number of word tokens in the entire collection, and $tf(w,\mathbf{D})$ is the frequency of word w in the collection $\mathbf{D}$. λ is the Dirichlet prior and is commonly set according to the average document length in the document collection.

Finally, researchers whose virtual documents have a high relevance score with the query will be returned as experts for a given query.

To make use of the profiling approaches to help expert finding, we first added the extracted profile into the virtual document of each researcher (+PE). We then employed the name disambiguation (+ND) method for expert finding. Name disambiguation can help filter some "experts" whose scores are accumulations of multiple researchers. Finally, we employed the user interest analysis result to help expert finding (+UIA). Specifically, we used the topic model to calculate another relevance score between the query term and the virtual document.

$$P_{ACT}(w|d,\theta,\phi) = \sum_{z=1}^{T} \sum_{x=1}^{A_d} P(w|z,\phi_z) P(z|x,\theta_x) P(x|d). \tag{24}$$

Then we combined the relevance score with that obtained using language model by multiplication:

$$P(w|d) = P_{LM}(w|d) \times P_{ACT}(w|d). \tag{25}$$

Finally, according to Equation (22), we can obtain a new score for each researcher and rank researchers based on the new relevance scores.

Figure 13 shows the results of expert finding. We see that significant improvements (+26%) can be obtained using the researcher profile information.

8. DEMONSTRATION SYSTEM

As a demonstration application, we have implemented the proposed profiling approaches in the academic search system: ArnetMiner.[8] In this system, we

[8] http://arnetminer.org

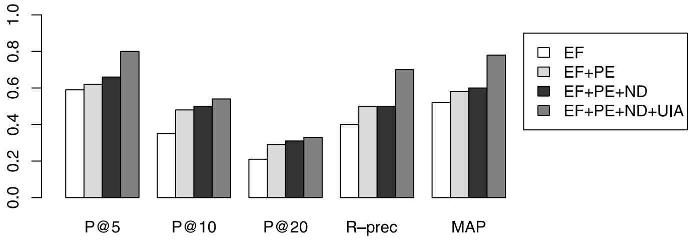


Fig. 13. Performances of expert finding.

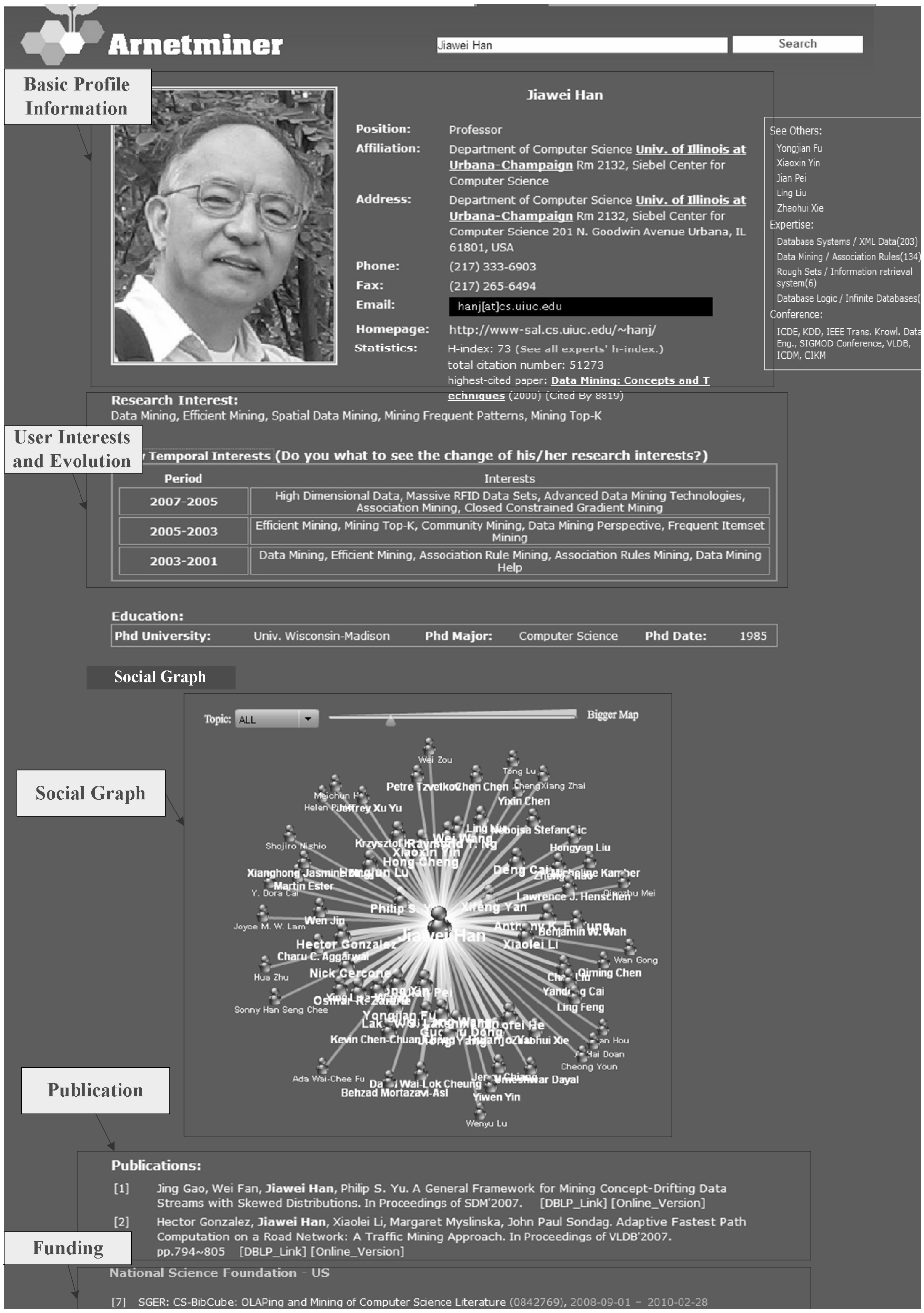
employed the proposed profiling approaches to extract profiles of researchers. So far, more than half a million researcher profiles and about three million publications have been extracted and integrated. The system has provided search services based on the profiling results, including expert finding, person search, association search, course search, etc.

Figure 14 shows an example researcher profile. We see that, at the top of the profile page, some basic information (e.g., person photo, position, and affiliation) of the researcher has been correctly extracted from the homepage. Below that is the research interest and evolution of the research interest discovered by our interest analysis approach. The bottom of the page lists publication papers of the researcher and closely above that is the social graph based on coauthor relationships.

The system has been in operation on the Internet for more than 3 years. System logs show that users of the system cover more than 190 countries. On average, the system receives about 2000 visits from independent IP addresses per day. The number of visits continuously increases by +20%/month. The top five countries where users come from the United States, China, Germany, India, and the United Kingdom.

We received feedback from thousands of users. Most of the feedback is positive. For example, some users have suggested that the profiling approach is useful and it can be enhanced by adding several new features. Some other feedback has also asked for improvements to the system. For example, 4.0% of the feedback complains about mistakes made in the profile extraction and 3.8% points out the integration mistakes (assigning publications to a wrong researcher). In addition, 3.5% of the feedback mentions that the found research interests are not accurate and the method should be improved, which is also our current research issue.

Further statistical analysis on thousands of extracted profiles and system logs shows that the profiling tasks are becoming more and more difficult. For example, we found that more than 70% of the researchers move at least one time. This immediately poses a challenging problem: how to maintain/update the extracted profiles. Moreover, the number of publications is increasing rapidly, particularly in recent years. Many research names that did not have the ambiguity problem now have to face that challenge. An interesting question also arises: can we make use of the time information to help disambiguation? Finally, the topics in the research community are also evolving



Basic Profile Information

Jiawei Han

Position: Professor
Affiliation: Department of Computer Science **Univ. of Illinois at Urbana-Champaign** Rm 2132, Siebel Center for Computer Science
Address: Department of Computer Science **Univ. of Illinois at Urbana-Champaign** Rm 2132, Siebel Center for Computer Science 201 N. Goodwin Avenue Urbana, IL 61801, USA
Phone: (217) 333-6903
Fax: (217) 265-6494
Email: hanj[at]cs.uiuc.edu
Homepage: http://www-sal.cs.uiuc.edu/~hanj/
Statistics: H-index: 73 (See all experts' h-index.)
total citation number: 51273
highest-cited paper: **Data Mining: Concepts and T echniques** (2000) (Cited By 8819)

See Others:
Yongjian Fu
Xiaoxin Yin
Jian Pei
Ling Liu
Zhaohui Xie
Expertise:
Database Systems / XML Data(203)
Data Mining / Association Rules(134)
Rough Sets / Information retrieval system(6)
Database Logic / Infinite Databases(
Conference:
ICDE, KDD, IEEE Trans. Knowl. Data Eng., SIGMOD Conference, VLDB, ICDM, CIKM

Research Interest:
Data Mining, Efficient Mining, Spatial Data Mining, Mining Frequent Patterns, Mining Top-K

User Interests and Evolution

Temporal Interests (Do you what to see the change of his/her research interests?)

Period	Interests
2007-2005	High Dimensional Data, Massive RFID Data Sets, Advanced Data Mining Technologies, Association Mining, Closed Constrained Gradient Mining
2005-2003	Efficient Mining, Mining Top-K, Community Mining, Data Mining Perspective, Frequent Itemset Mining
2003-2001	Data Mining, Efficient Mining, Association Rule Mining, Association Rules Mining, Data Mining Help

Education:

Phd University:	Univ. Wisconsin-Madison	**Phd Major:**	Computer Science	**Phd Date:**	1985

Social Graph

Social Graph

Topic: ALL

Publication

Publications:

[1] Jing Gao, Wei Fan, **Jiawei Han**, Philip S. Yu. A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributions. In Proceedings of SDM'2007. [DBLP_Link] [Online_Version]

[2] Hector Gonzalez, **Jiawei Han**, Xiaolei Li, Margaret Myslinska, John Paul Sondag. Adaptive Fastest Path Computation on a Road Network: A Traffic Mining Approach. In Proceedings of VLDB'2007. pp.794~805 [DBLP_Link] [Online_Version]

Funding

National Science Foundation - US

[7] SGER: CS-BibCube: OLAPing and Mining of Computer Science Literature (0842769), 2008-09-01 – 2010-02-28

Fig. 14. An example researcher profile.

quickly. It is necessary to capture the evolution pattern for user interest analysis?

So far, the system and the proposed approaches mainly focus on academic data. For extending the techniques to the problem of general Web user profiling, we need to consider many other problems. For example, on the social Web, the data may contain much noise. The first problem we should address before extraction and mining is how to filter the noisy data and how to normalize the informal text into a standard format [Zhu et al. 2007].

9. RELATED WORK

9.1 User Profiling

There are two types of research work on user profiling: profile extraction and profile learning.

Several research efforts have been made for extracting profile information about a person. For example, Yu et al. [2005] proposed a cascaded information extraction framework for identifying personal information from resumes. In the first pass, a resume is segmented into consecutive blocks attached with labels indicating the information type. And in the second pass, detailed information such as address and email are identified in certain blocks. The Artequakt system [Alani et al. 2003] uses a rule-based extraction system called GATE [Cunningham et al. 2002] to extract entity and relation information from the Web. Michelson and Knoblock [2007] proposed an unsupervised method to extract information from the Web. However, most of previous work viewed the profile extraction as several separate issues conducted in a more-or-less ad hoc manner.

A few efforts also have been made to extract contact information from email or the Web. For example, Kristjansson et al. [2004] developed an interactive information extraction system to assist the user to populate a contact database from emails. See also Balog et al. [2006]. Contact information extraction is a subtask of profile extraction; thus it differs significantly from the profile extraction.

Many information extraction models have been proposed. Hidden Markov Model (HMM) [Ghahramani and Jordan 1997], Maximum Entropy Markov Model (MEMM) [McCallum et al. 2000], Conditional Random Field (CRF) [Lafferty et al. 2001], Support Vector Machines (SVM) [Cortes and Vapnik 1995], and Voted Perceptron [Collins 2002] are widely used models. Sarawagi and Cohen [2004] also proposed a semi-Markov Conditional Random Fields model for information extraction. However, most of the existing models do not consider the hierarchically laid-out structure on the Web. Tang et al. [2007] gave an overview of the existing literatures on information extraction.

The other type of research is to learn the user profile from user associated documents or user visiting logs. For example, Pazzani and Billsus [1997] discussed algorithms for learning and revising user profiles that can determine which World Wide Web sites on a given topic would be interesting to a user.

It uses a naive Bayes classifier to incrementally learn profiles from user feedback on the Web sites. [Chan 1999] has developed a personalized Web browser. It learns a user profile, and aims at helping users navigating the Web by searching for potentially interesting pages for recommendations. Soltysiak and Crabtree [1998] described an experimental work to study whether user interests can be automatically classified through heuristics. The results highlighted the need for user feedback and machine learning methods.

9.2 Name Disambiguation

A number of approaches have been proposed for name disambiguation in different domains.

For example, Bekkerman and McCallum [2005] tried to distinguish Web pages involving different individuals with the same name. They presented two unsupervised frameworks for solving this problem: one based on the link structure of the Web pages and the other on the agglomerative/conglomerative clustering method. The methods are based on unsupervised clustering and cannot describe the relationships between data points.

There are also many works focusing on name disambiguation on publication data. For example, Han et al. [2005] proposed an unsupervised learning approach using a K-way spectral clustering method. They calculate a Gram matrix for each name dataset and apply the K-way spectral clustering algorithm to the Gram matrix to get the results. On and Lee [2007] proposed a scalable algorithm for the name disambiguation problem. They adapted the multilevel graph partition technique to solve the large-scale name disambiguation problem. Their algorithm can have a magnitude improvement in terms of efficiency. Bhattacharya and Getoor [2007] proposed a relational clustering algorithm that uses both attribute and relational information for disambiguation. See also Tan et al. [2006]. This type of method usually uses a parameter-fixed distance metric in the clustering algorithm, while parameters of the distance metric can be learned during the disambiguation.

Two supervised methods were proposed in Han et al. [2004] based on naive Bayes and Support Vector Machines. For a given author name, the methods learn a specific model from the train data and use the model to predict whether a new paper was authored by an author with the name. However, the method is user dependent. It is impractical to train thousands of models for all individuals in a large digital library. In contrast to supervised methods, our method has more scalability.

The other type of related work is semisupervised clustering [Basu et al. 2004; Cohn et al. 2003; Zhang et al. 2007a]. Basu et al. [2004] proposed a probabilistic model for semisupervised clustering based on Hidden Markov Random Fields. Their model combines the constraint-based and distance-based approaches.

9.3 Topic Modeling

Much effort has gone into investigating topic model or latent semantic structure discovery.

Probabilistic latent semantic indexing (pLSI) was proposed in Hofmann [1999]. The difference between LSA and pLSI is that the latter is based on the likelihood principle and defines a proper generative model of the data; hence it results in a more solid statistical foundation. However, pLSI has the problem of overfitting and not being able to estimate documents outside of the training set.

Blei et al. [2003] introduced a new semantically consistent topic model, Latent Dirichlet Allocation (LDA). The basic generative process of LDA closely resembles pLSI except that, in pLSI, the topic mixture is conditioned on each document and, in LDA, the topic mixture is drawn from a conjugate Dirichlet prior, which remains the same for all documents.

Some other work has also aimed at modeling author interests and document content simultaneously. For example, the Author model (also termed the *Multi-label Mixture Model*) [McCallum 1999] is aimed at modeling the author interests with a one-to-one correspondence between topics and authors. Rosen-Zvi et al. [2004] presented an Author-Topic model, which integrates authorship into the topic model and thus can be used to find a topic distribution over documents and a mixture of the distributions associated with authors.

McCallum et al. [2007] have studied several other topic models in social network analysis. They proposed the Author-Recipient-Topic (ART) model, which learns topic distributions based on emails sent between people.

Compared with above topic modeling work, in this article, we aim at using a unified model (author-conference-topic model) to characterize the topic distributions of multiple interdependent objects in the academic social network.

10. CONCLUSION AND FUTURE WORK

In this article, we have investigated the problem of Web user profiling. We have formalized the profiling problem as several subtasks. We have proposed a combination approach to deal with the problems. Specifically, we have proposed a Tree-structured Conditional Random Field (TCRF) to extract the profile information from the Web page and proposed a probabilistic model to solve the name ambiguity problem for integrating the profile information from different sources. Further, we have proposed a topic model to discover user interests. Experimental results indicate that our proposed methods outperform the baseline methods. Experiments on expert finding also show that the extracted user profiles can be used to improve the accuracy of expert finding. We have developed a demonstration system based on the proposed approaches. User feedback and system logs show that users of the system consider the system useful.

There are several potential enhancements of this work. First, a general Web page may contain a lot of noise, and how to extract accurate profile information from the noisy data is a challenging issue. Second, the performance of name disambiguation can be further improved by incorporating other relationships or human background knowledge. Third, the proposed method for user interest discovery is an unsupervised method and does not consider any domain knowledge. In practice, for a specific domain (e.g., computer science), people may already build some taxonomy (e.g., the ACM categories) to describe the subfields in the domain, which can be used to guide the discovery of user interests.

There are also many other future directions for this work. It would be interesting to investigate how to extract the profile based on partially labeled data. Data labeling for machine learning is usually tedious and time consuming. How to reduce the labeling work is a challenging problem. It would also be interesting to investigate the dynamic problem. The profile of a researcher might change after years, for example, by the individual moving to a new company. Furthermore, in-depth analysis of the user profiles is also important.

REFERENCES

ALANI, H., KIM, S., MILLARD, D. E., WEAL, M. J., HALL, W., LEWIS, P. H., AND SHADBOLT, N. R. 2003. Automatic ontology-based knowledge extraction from web documents. *IEEE Intell. Syst. 18*, 1, 14–21.

ANDRIEU, C., DE FREITAS, N., DOUCET, A., AND JORDAN, M. I. 2003. An introduction to MCMC for machine learning. *Mach. Learn. 50*, 5–43.

BAEZA-YATES, R. AND RIBEIRO-NETO, B. 1999. *Modern Information Retrieval*. ACM Press, New York, NY.

BALOG, K., AZZOPARDI, L., AND DE RIJKE, M. 2006. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th ACM SIGIR International Conference on Information Retrieval (SIGIR)*. 43–55.

BASU, S., BILENKO, M., AND MOONEY, R. J. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 59–68.

BEKKERMAN, R. AND MCCALLUM, A. 2005. Disambiguating web appearances of people in a social network. In *Proceedings of the 14th International Conference on World Wide Web (WWW)*. 463–470.

BHATTACHARYA, I. AND GETOOR, L. 2007. Collective entity resolution in relational data. *ACM Trans. Knowl. Disc. Data 1*, 1, 1–36.

BLEI, D. M., NG, A. Y., AND JORDAN, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res. 3*, 993–1022.

BRICKLEY, D. AND MILLER, L. 2004. Foaf vocabulary specification. In *Namespace Document.* http://xmlns.com/foaf/0.1/.

BUCKLEY, C. AND VOORHEES, E. M. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 25–32.

CAI, D., HE, X., AND HAN, J. 2007. Spectral regression for dimensionality reduction. Tech. rep. UIUCDCS-R-2007-2856, University of Illinois at Urbana-Champaign, Urbana, IL.

CAO, Y., LI, H., AND LI, S. 2003. Learning and exploiting non-consecutive string patterns for information extraction. Tech. rep. MSR-TR-2003-33. Microsoft Research, Redmond, WA.

CHAN, P. K. 1999. A non-invasive learning approach to building web user profiles. In *Proceedings of the Workshop on Web Usage Analysis and User Profiling (KDD'99)*.

CIRAVEGNA, F. 2001. An adaptive algorithm for information extraction from web-related texts. In *Proceedings of the IJCAI Workshop on Adaptive Text Extraction and Mining*.

COHN, D., CARUANA, R., AND MCCALLUM, A. 2003. Semi-supervised clustering with user feedback. Tech. rep. TR2003-1892, Cornell University, Ithaca, NY.

COLLINS, M. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1–8.

CORTES, C. AND VAPNIK, V. 1995. Support-vector networks. *Mach. Learn. 20*, 273–297.

CRASWELL, N., DE VRIES, A. P., AND SOBOROFF, I. 2005. Overview of the trec-2005 enterprise track. In *TREC Conference Notebook*. 199–205.

CUNNINGHAM, H., MAYNARD, D., BONTCHEVA, K., AND TABLAN, V. 2002. Gate: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

GHAHRAMANI, Z. AND JORDAN, M. I. 1997. Factorial hidden markov models. *Mach. Learn. 29,* 2-3, 245–273.

GRIFFITHS, T. L. AND STEYVERS, M. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences (PNAS)*. 5228–5235.

HAMMERSLEY, J. M. AND CLIFFORD, P. 1971. Markov field on finite graphs and lattices. *Unpublished manuscript*.

HAN, H., GILES, L., ZHA, H., LI, C., AND TSIOUTSIOULIKLIS, K. 2004. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital libraries (JCDL)*. 296–305.

HAN, H., ZHA, H., AND GILES, C. L. 2005. Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital libraries (JCDL)*. 334–343.

HOFMANN, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR)*. 50–57.

KRISTJANSSON, T., CULOTTA, A., VIOLA, P., AND MCCALLUM, A. 2004. Interactive information extraction with constrained conditional random fields. In *Proceedings of AAAI*.

LAFFERTY, J. D., MCCALLUM, A., AND PEREIRA, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*. 282–289.

LIU, D. C., NOCEDAL, J., AND C, D. 1989. On the limited memory BFGS method for large scale optimization. *Math. Program. 45*, 503–528.

MCCALLUM, A. 1999. Multi-label text classification with a mixture model trained by EM. In *Proceedings of the AAAI Workshop on Text Learning*.

MCCALLUM, A., FREITAG, D., AND PEREIRA, F. C. N. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*. 591–598.

MCCALLUM, A., WANG, X., AND CORRADA-EMMANUEL, A. 2007. Topic and role discovery in social networks with experiments on Enron and academic email. *J. Art. Intell. Res. 30*, 249–272.

MICHELSON, M. AND KNOBLOCK, C. 2007. Unsupervised information extraction from unstructured, ungrammatical data sources on the World Wide Web. *Int. J. Doc. Anal. Recog. 10*, 3, 211–266.

MIMNO, D. AND MCCALLUM, A. 2007. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 500–509.

MINKA, T. 2003. Estimating a dirichlet distribution. Tech. rep.
`http://research.microsoft.com/ minka/papers/dirichlet/`.

NEWMAN, D., ASUNCION, A., SMYTH, P., AND WELLING, M. 2007. Distributed inference for latent dirichlet allocation. In *Proceedings of the 19th Neural Information Processing Systems (NIPS)*.

ON, B.-W. AND LEE, D. 2007. Scalable name disambiguation using multi-level graph partition. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*.

PAZZANI, M. J. AND BILLSUS, D. 1997. Learning and revising user profiles: The identification of interesting web sites. *Mach. Learn. 27,* 3, 313–331.

ROSEN-ZVI, M., GRIFFITHS, T., STEYVERS, M., AND SMYTH, P. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th International Conference on Uncertainty in Artificial Intelligence (UAI)*. 487–494.

SARAWAGI, S. AND COHEN, W. W. 2004. Semi-Markov conditional random fields for information extraction. In *Proceedings of the 17th Neural Information Processing Systems (NIPS)*. 1185–1192.

SHA, F. AND PEREIRA, F. 2003. Shallow parsing with conditional random fields. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL)*. 134–141.

SOLTYSIAK, S. J. AND CRABTREE, I. B. 1998. Automatic learning of user profiles—towards the personalisation of agent services. *BT Tech. J. 16,* 3, 110–117.

SPROAT, R., BLACK, A. W., CHEN, S., KUMAR, S., OSTENDORF, M., AND RICHARDS, C. 2001. Normalization of non-standard words. *Comput. Speech Lang.*, 287–333.

STEYVERS, M., SMYTH, P., AND GRIFFITHS, T. 2004. Probabilistic author-topic models for information discovery. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 306–315.

TAN, Y. F., KAN, M.-Y., AND LEE, D. 2006. Search engine driven author disambiguation. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital libraries (JCDL)*. 314–315.

TANG, J., HONG, M., LI, J., AND LIANG, B. 2006. Tree-structured conditional random fields for semantic annotation. In *Proceedings of the 5th International Semantic Web Conference (ISWC)*. 640–653.

TANG, J., HONG, M., ZHANG, D., LIANG, B., AND LI, J. 2007. Information extraction: Methodologies and applications. In *Emerging Technologies of Text Mining: Techniques and Applications*. H. A. Prado and E. Ferneda, Eds. Idea Group Inc., Hershey, PA.

TANG, J., JIN, R., AND ZHANG, J. 2008a. A topic modeling approach and its integration into the random walk framework for academic search. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*. 1055–1060.

TANG, J., ZHANG, J., YAO, L., LI, J., ZHANG, L., AND SU, Z. 2008b. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 990–998.

TEH, Y. W., JORDAN, M. I., BEAL, M. J., AND BLEI, D. M. 2004. Hierarhical dirichlet processes. Tech. rep. 653. Department of Statistics, University of California, Berkeley, CA.

VAN RIJSBERGEN, C. 1979. *Information Retrieval*. Butterworths, London, U.K.

WAINWRIGHT, M. J., JAAKKOLA, T., AND WILLSKY, A. S. 2001. Tree-based reparameterization for approximate estimation on loopy graphs. In *Proceedings of the 13th Neural Information Processing Systems (NIPS)*. 1001–1008.

XUN, E., HUANG, C., AND ZHOU, M. 2000. A unified statistical model for the identification of English BASENP. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*. 3–6.

YEDIDIA, J. S., FREEMAN, W. T., AND WEISS, Y. 2001. Generalized belief propagation. In *Proceedings of the 13th Neural Information Processing Systems (NIPS)*. 689–695.

YIN, X., HAN, J., AND YU, P. 2007. Object distinction: Distinguishing objects with identical names. In *Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE)*. 1242–1246.

YU, K., GUAN, G., AND ZHOU, M. 2005. Resume information extraction with cascaded hybrid model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*. 499–506.

ZHAI, C. AND LAFFERTY, J. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th ACM SIGIR International Conference on Information Retrieval (SIGIR)*. 334–342.

ZHANG, D., TANG, J., AND LI, J. 2007a. A constraint-based probabilistic framework for name disambiguation. In *Proceedings of the 16th Conference on Information and Knowledge Management (CIKM)*. 1019–1022.

ZHANG, J., TANG, J., AND LI, J. 2007b. Expert finding in a social network. In *Proceedings of the 12th Database Systems Conference for Advanced Applications (DASFAA),* 1066–1069.

ZHU, C., TANG, J., LI, H., NG, H. T., AND ZHAO, T. 2007. A unified tagging approach to text normalization. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*. 689–695.

ZHU, J., NIE, Z., WEN, J.-R., ZHANG, B., AND MA, W.-Y. 2006. Simultaneous record detection and attribute labeling in Web data extraction. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 494–503.

Received December 2009; accepted February 2010