ArnetMiner: Extraction and Mining of Academic Social Networks

Jie Tang, Jing Zhang
Computer Science Dept.
Tsinghua University, China
jietang@tsinghua.edu.cn
zhangjing0544@gmail.com

Limin Yao, Juanzi Li
Computer Science Dept.
Tsinghua University, China
ylm@keg.cs.tsinghua.edu.cn
ljz@keg.cs.tsinghua.edu.cn

Li Zhang, Zhong Su
IBM, China Research Lab
Beijing, China
lizhang@cn.ibm.com
suzhong@cn.ibm.com

ABSTRACT

This paper addresses several key issues in the ArnetMiner system, which aims at extracting and mining academic social networks. Specifically, the system focuses on: 1) Extracting researcher profiles automatically from the Web; 2) Integrating the publication data into the network from existing digital libraries; 3) Modeling the entire academic network; and 4) Providing search services for the academic network. So far, 448,470 researcher profiles have been extracted using a unified tagging approach. We integrate publications from online Web databases and propose a probabilistic framework to deal with the name ambiguity problem. Furthermore, we propose a unified modeling approach to simultaneously model topical aspects of papers, authors, and publication venues. Search services such as expertise search and people association search have been provided based on the modeling results. In this paper, we describe the architecture and main features of the system. We also present the empirical evaluation of the proposed methods.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Text Mining, Digital Libraries; H.2.8 [**Database Management**]: Database Applications

General Terms

Algorithms, Experimentation

Keywords

Social Network, Information Extraction, Name Disambiguation, Topic Modeling, Expertise Search, Association Search

1. INTRODUCTION

Extraction and mining of academic social networks aims at providing comprehensive services in the scientific research field. In an academic social network, people are not only interested in searching for different types of information (such as authors, conferences, and papers), but are also interested in finding semantics-based information (such as structured researcher profiles).

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.

Many issues in academic social networks have been investigated and several systems have been developed (e.g., DBLP, CiteSeer, and Google Scholar). However, the issues were usually studied separately and the methods proposed are not sufficient for mining the entire academic network. Two reasons are as follows: 1) Lack of semantics-based information. The social information obtained from user-entered profiles or by extraction using heuristics is sometimes incomplete or inconsistent; 2) Lack of a unified approach to efficiently model the academic network. Previously, different types of information in the academic network were modeled individually, thus dependencies between them cannot be captured accurately.

In this paper, we try to address the two challenges in novel approaches. We have developed an academic search system, called ArnetMiner (`http://www.arnetminer.org`). Our objective in this system is to answer the following questions: 1) how to automatically extract researcher profiles from the Web? 2) how to integrate the extracted information (e.g., researchers' profiles and publications) from different sources? 3) how to model different types of information in a unified approach? and 4) how to provide powerful search services based on the constructed network?

(1) We extend the Friend-Of-A-Friend (FOAF) ontology [9] as the profile schema and propose a unified approach based on Conditional Random Fields to extract researcher profiles from the Web.

(2) We integrate the extracted researcher profiles and the crawled publication data from the online digital libraries. We propose a unified probabilistic framework for dealing with the name ambiguity problem in the integration.

(3) We propose three generative probabilistic models for simultaneously modeling topical aspects of papers, authors, and publication venues.

(4) Based on the modeling results, we implement several search services such as expertise search and association search.

We conducted empirical evaluations of the proposed methods. Experimental results show that our proposed methods significantly outperform the baseline methods for dealing with the above issues.

Our contributions in this paper include: (1) a proposal of a unified tagging approach to researcher profile extraction, (2) a proposal of a unified probabilistic framework to name disambiguation, and (3) a proposal of three probabilistic topic models to simultaneously model the different types of information.

The paper is organized as follows. In Section 2, we review the related work. In Section 3, we give an overview of the system. In Section 4, we present our approach to researcher profiling. In Section 5, we describe the probabilistic framework to name disambiguation. In Section 6, we propose three generative probabilistic models to model the academic network. Section 7 illustrates several search services provided in ArnetMiner based on the modeling results. We conclude the paper in Section 8.

2. RELATED WORK

2.1 Person Profile Extraction

Several research efforts have been made for extracting person profiles. For example, Yu et al. [32] propose a two-stage extraction method for identifying personal information from resumes. The first stage segments a resume into different types of blocks and the second stage extracts the detailed information such as Address and Email from the identified blocks. However, the method formalizes the profile extraction as several separate steps and conducts extraction in a more or less ad-hoc manner.

A few efforts also have been placed on the extraction of contact information from emails or from the Web. For example, Kristjansson et al. [19] have developed an interactive information extraction system to assist the user to populate a contact database from emails. In comparison, profile extraction consists of contact information extraction as well as other different subtasks.

2.2 Name Disambiguation

A number of approaches have been proposed to name disambiguation. For example, Bekkerman and McCallum [6] present two unsupervised methods to distinguish Web pages to different persons with the same name: one is based on the link structure of the Web pages and the other is based on the textural content. However, the methods cannot incorporate the relationships between data.

Han et al. [15] propose an unsupervised learning approach using K-way spectral clustering. Tan et al. [27] propose a method for name disambiguation based on hierarchical clustering. However, this kind of methods cannot capture the relationships either.

Two supervised methods are proposed by Han et al. [14]. For each given name, the methods learn a specific classification model from the training data and use the model to predict whether a new paper is authored by a specific author with the name. However, the methods are user-dependent. It is impractical to train thousands of models for all individuals in a large digital library.

2.3 Topic Modeling

Considerable work has been conducted for investigating topic models or latent semantic structures for text mining. For example, Hofmann [17] proposes the probabilistic latent semantic indexing (pLSI) and applies it to information retrieval (IR).

Blei et al. [8] introduce a three-level Bayesian network, called 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 while in LDA, the topic mixture is drawn from a conjugate Dirichlet prior that remains the same for all documents.

Some other work has been conducted for modeling both author interests and document contents together. For example, the Author model [21] is aimed at modeling the author interests with a one-to-one correspondence between topics and authors. The Author-Topic model [25] [26] integrates the authorship into the topic model and can find a topic mixture over documents and authors.

Compared with the previous topic modeling work, in this paper, we propose a unified topic model to simultaneously model the topical aspects of different types of information in the academic network.

2.4 Academic Search

For academic search, several research issues have been intensively investigated, for example expert finding and association search.

Expert finding is one of the most important issues for mining social networks. For example, both Nie et al. [24] and Balog et al. [4] propose extended language models to address the expert finding problem. From 2005, Text REtrieval Conference (TREC) has provided a platform with the Enterprise Search Track for researchers to empirically assess their methods for expert finding [13].

Association search aims at finding connections between people. For example, the ReferralWeb [18] system helps people search and explore social networks on the Web. Adamic and Adar [1] have investigated the problem of association search in email networks. However, existing work mainly focuses on how to find connections between people and ignores how to rank the found associations.

In addition, a few systems have been developed for academic search such as, scholar.google.com, libra.msra.cn, citeseer.ist.psu, and Rexa.info. Though much work has been performed, to the best of our knowledge, the issues we focus on in this work (i.e., profile extraction, name disambiguation, and academic network modeling) have not been sufficiently investigated. Our system addresses all these problems holistically.

3. OVERVIEW OF ARNETMINER

Figure 1 shows the architecture of our ArnetMiner system. The system mainly consists of five main components:

1. *Extraction*: it focuses on extracting researcher profiles from the Web automatically. It first collects and identifies one's homepage from the Web, then uses a unified approach to extract the profile properties from the identified document. It extracts publications from online digital libraries using rules.
2. *Integration*: it integrates the extracted researchers' profiles and the extracted publications by using the researcher name as the identifier. A probabilistic framework has been proposed to deal with the name ambiguity problem in the integration. The integrated data is stored into a researcher network knowledge base (RNKB).
3. *Storage and Access*: it provides storage and index for the extracted/integrated data in the RNKB. Specifically, for storage it employs MySQL and for index, it employs the inverted file indexing method [3].
4. *Modeling*: it utilizes a generative probabilistic model to simultaneously model different types of information. It estimates a topic distribution for each type of information.
5. *Search Services*: based on the modeling results, it provides several search services: expertise search and association search. It also provides other services, e.g., author interest finding and academic suggestion (such as paper suggestion and citation suggestion).

It is challenging in many ways to implement these components. First, the previous extraction work has been usually conducted on a specific data set. It is not immediately clear whether such methods can be directly adapted to the global Web. Secondly, it is unclear how to deal with the disambiguation problem by making full use of the extracted information. For example, how to use the relationships between publications. Thirdly, there is no existing model that can simultaneously model the different types of information in the academic network. Finally, different strategies for modeling the academic network have different behaviors. It is necessary to study how different they are and which one would be the best for academic search.

Based on these considerations, for profile extraction, name disambiguation, and modeling, we propose new approaches to overcome the drawbacks that exist in the traditional methods. For storage and access, we utilize the classical methods, because these issues have been intensively investigated and the existing methods can result in good performance in our system.

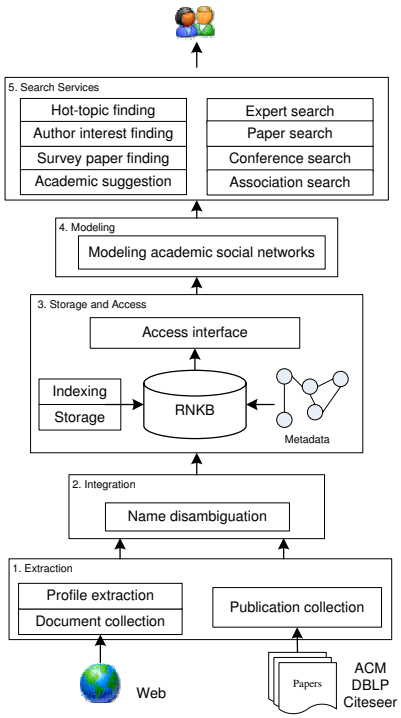


Figure 1: Architecture of ArnetMiner.

4. RESEARCHER PROFILE EXTRACTION

4.1 Problem Definition

Profile extraction is the process of extacting the value of each property in a person profile. We define the schema of the researcher profile (as shown in Figure 2) by extending the FOAF ontology [9].

We perform a statistical study on randomly selected 1,000 researchers from ArnetMiner and find that it is non-trivial to perform profile extraction from the Web. We observed that 85.62% of the researchers are faculty members from universities and 14.38% are from company research centers. For researchers from the same company, they may share a template-based homepage. However, different companies have different templates. For researchers from universities, the layout and the content of their homepages vary largely. We have also found that 71.88% of the 1,000 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 also analyze the content of the Web pages and find that about 40% of the profile properties are presented in tables/lists and the others are presented in natural language text. This suggests a method without using 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 (14.54%) in our data of which the extraction needs to use the extraction results of other properties. An ideal method should consider processing all the subtasks holistically.

4.2 A Unified Approach to Profiling

4.2.1 Process

The proposed approach consists of three steps: relevant page identification, preprocessing, and extraction. In relevant page identification, given a researcher name, we first get a list of web pages by a search engine (we use the Google API) and then identify the homepage/introducing page using a binary classifier. We use Support Vector Machines (SVM) [12] 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 F1-measure. In preprocessing, (a) we separate 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 in the tagging problem. In tagging, given a sequence of units, we determine the most likely corresponding sequence of tags by using a trained tagging model. Each tag corresponds to a property defined in Figure 2, e.g., 'Position'. In this paper, we make use of Conditional Random Fields (CRFs) [20] as the tagging model. Next we describe the steps (a) and (b) in detail.

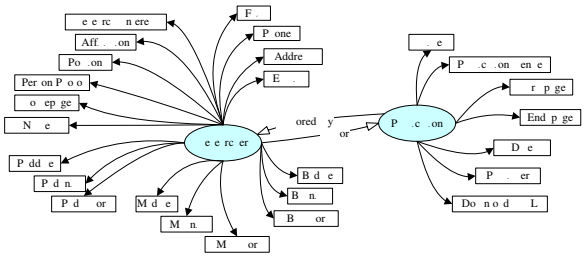


Figure 2: The schema of researcher profile.

(a) We identify tokens in the Web page using heuristics. 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 include email, URL, date, number, percentage, words containing special terms (e.g. 'Ph.D.' and '.NET'), special symbols (e.g. '===' and '###'). We identify special words by using regular expressions. '<image>' tokens (used for identifying person photos and email addresses) are '<image>' tags in the HTML file. Terms are base noun phrases extracted from the Web page by using a tool based on technologies proposed in [30].

(b) We assign tags to each token based on the token type. For example, for a standard word, we assign all possible tags corresponding to all properties. For a special word, we assign tags indicating Position, Affiliation, Email, Address, Phone, Fax, Bsdate, Msdate, and Phddate. For a '<image>' token, we assign two tags: Photo and Email, because an email address is sometimes shown as an image).

After each token is assigned with several possible tags, we can perform most of the profiling tasks using the tags (extracting 19 properties defined in Figure 2).

4.2.2 CRF model and Features

We employ Conditional Random Fields (CRF) as the tagging model. CRF is a conditional probability of a sequence of tags given a sequence of observations [20]. For tagging, a trained CRF model is used to find the sequence of tags Y^* having the highest likelihood $Y^* = max_Y P(Y|X)$. The CRF model is built with the labeled data by means of an iterative algorithm based on Maximum Likelihood Estimation.

Three types of features were defined in the CRF model: content features, pattern features, and term features. The features were defined for different kinds of tokens. Table 1 shows the defined features. We incorporate the defined features into the CRF model by defining Boolean-valued feature functions. Finally, 108,409 features were used in our experiments.

4.3 Profile Extraction Performance

For evaluating our profiling method, we randomly chose 1,000 researcher names in total from our researcher network. We used the

Table 1: Content features, Pattern features, and term features.

Content Feature			Pattern Feature	
Standard Token	Word	Word in the token	**All Token**	
	Morphology	Morphology of the word	Positive word	If the token contains a pre-defined positive word
Image Token	Size	The size of the image	Negative word	If the token contains a pre-defined negative word
	Height/width ratio	the ratio of height/width of the image	Special token	If the token contains a special pattern
	Image format	The format of the image (e.g., 'JPG')	Name	If the token contains the researcher name
	Image color	The number of unique colors	#Line break	How many line breaks before the current line
		The number of bits per pixel	**Term Feature**	
	Filename	Words in the filename	**Term Token**	
	Face detection	If the image contains a person face recognized by (opencvlibrary.sf.net)	Term	If the token contains a base noun phrase
	ALT	Words in 'alt' attribute of the image	Dictionary	If the token contains a word in a dictionary

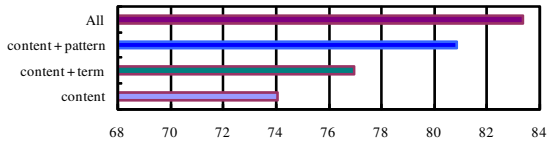


Figure 3: Contribution of features (%).

method described in Section 4.2.1 to find the researchers' homepages or introducing pages. If the method cannot find a Web page for a researcher, we removed the researcher name from the data set. We finally obtained 898 Web pages (one for each researcher). Seven human annotators conducted annotation on the Web pages. A spec was created to guide the annotation process. On disagreements in the annotation, we conducted 'majority voting'. In the experiments, we conducted evaluations in terms of precision, recall, and F1-measure for each profile property.

We defined baselines for profile extraction. We used the rule learning and the classification based approaches as baselines. For the former, we employed the Amilcare tool, which is based on a rule induction algorithm: LP^2 [11]. For the latter, we trained a classifier to identify the value of each property. We employed Support Vector Machines (SVM) [12] as the classification model.

Experimental results show that our method results in a performance of 83.37% in terms of average F1-measure; while Amilcare and SVM result in 53.44% and 73.57%, respectively. Our method clearly outperforms the two baseline methods. We have also found that the performance of the unified method decreases (-11.28% by F1) when removing the transition features, which indicates that a unified approach is necessary for researcher profiling.

We investigated the contribution of each feature type in profile extraction. We employed only content features, content+term features, content+pattern features, and all features to train the models and conducted the profile extraction. Figure 3 shows the average F1-scores of profile extraction with different feature types. The results unveil contributions of individual features in the extraction. We see that solely using one type of features cannot obtain accurate profiling results. Detailed evaluations can be found in [28].

5. NAME DISAMBIGUATION

5.1 Problem Definition

We integrate the publication data from the online database including DBLP bibliography, ACM Digital library, CiteSeer, and others. For integrating the researcher profiles and the publications, we use the researcher name and the publication author name as the identifier. The method inevitably has the ambiguity problem.

We give a formal definition of the name disambiguation task in our context. Given a person name a, we denote all publications having the author name a as $P = \{p_1, p_2, \cdots, p_n\}$. Each publication p_i has six attributes: paper title ($p_i.title$), publication venue ($p_i.pubvenue$), published year ($p_i.year$), abstract ($p_i.abstract$), authors ($\{a_i^{(0)}, a_i^{(1)}, \cdots, a_i^{(u)}\}$), and references ($p_i.references$).

Table 2: Relationships between papers.

R	W	Relation Name	Description
r_1	w_1	CoPubvenue	p_i:pubvenue $= p_j$:pubvenue
r_2	w_2	CoAuthor	$9r; s > 0; a_i^{(r)} = a_j^{(s)}$
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	¿-CoAuthor	¿-extension co-authorship (¿ > 1)

For the authors of a paper $\{a_i^{(0)}, a_i^{(1)}, \cdots, a_i^{(u)}\}$, we call the author name we are going to disambiguate as 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 author a to their actual researcher y_h, $h \in [1, k]$.

We define five types of relationships between papers (Table 2). Relationship r_1 represents two papers are published at the same venue. Relationship r_2 means two papers have a secondary author with the same name, and relationship r_3 means one paper cites the other paper. Relationship r_4 indicates a constraint-based relationship supplied via user feedback. For instance, the user can specify that two specific papers should be assigned to a same person. 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 to disambiguate 'David Mitchell'. If 'Andrew Mark' and 'Fernando Mulford' also coauthor a paper, then we say p_i and p_j have a 2-CoAuthor relationship. In our currently experiments, we empirically set the weights of relationships $w_1 \sim w_5$ as 0.2, 0.7, 0.3, 1.0, 0.7^{τ}.

The publication data with relationships can be modeled as a graph comprising of nodes and edges. Each attribute of a paper is attached to the corresponding node as a feature vector. In the vector, we use words (after stop words filtering and stemming) in the attributes as features and use their numbers of occurrences as the values.

5.2 A Unified Probabilistic Framework

5.2.1 *Formalization using HMRF*

We propose a probabilistic framework based on Hidden Markov Random Fields (HMRF) [5], which can capture dependencies between observations (with each paper being viewed as an observation). The disambiguation problem is cast as assigning a tag to each paper with each tag representing an actual researcher.

Specifically, we define a-posteriori probability as the objective function. We aims at maximizing the objective function. The five types of relationships are incorporated into the objective function. According to HMRF, the conditional distribution of the researcher labels y given the observations x (papers) is

$$P(y|x) = \frac{1}{Z} exp(\sum_{i,h} D(x_i; y_h) \sum_{i,j \neq i} (D(x_i; x_j) \sum_{r_k} w_k r_k(x_i; x_j)))$$

Table 3: Data set for name disambiguation.

Person Name	# Publi-cations	#Actual Persons	Person Name	# Publi-cations	#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					

Recently, probabilistic topic models such as probabilistic Latent Semantic Indexing (pLSI) [17], Latent Dirichlet Allocation (LDA) [8], and Author-Topic model [25] [26] have been proposed as well as successfully applied to multiple text mining tasks such as information retrieval [29], collaborative filtering [8] [16], and paper reviewer finding [22]. However, these models are not sufficient to model the whole academic network, as they cannot model topical aspects of all types of information in the academic network.

We propose a unified topic model for simultaneously modeling the topical distribution of papers, authors, and conferences. For simplicity, we use conference to denote conference, journal, and book hereafter. The learned topic distribution can be used to further estimate the inter-dependencies between different types of information, e.g., the closeness between a conference and an author.

The notations used are summarized as follows. A paper d is a vector $\mathbf{w}_d$ of N_d words, in which each w_{di} is chosen from a vocabulary of size V; a vector $\mathbf{a}_d$ of A_d authors, chosen from a set of authors of size A; and a published conference c_d. A collection of D papers is defined by $\mathbf{D} = \{(\mathbf{w}_1, \mathbf{a}_1, c_1), \cdots, (\mathbf{w}_D, \mathbf{a}_D, c_D)\}$. x_{di} denotes an author, chosen from $\mathbf{a}_d$, responsible for the i-th word w_{di} in paper d. The number of topics is denoted as T.

6.1 Our Proposed Topic Models

The proposed model is called Author-Conference-Topic (ACT) model. Three different strategies are employed to implement the topic model (as shown in Figure 6).

In the first model (ACT1, Figure 6 (a)), each author is associated with a multinomial distribution over topics and each word in a paper and the conference stamp is generated from a sampled topic.

In the second model (ACT2, Figure 6 (b)), each author-conference pair is associated with a multinomial distribution over topics and each word is then generated from a sampled topic.

In the third model (ACT3, Figure 6 (c)), each author is associated with a topic distribution and the conference stamp is generated after topics have been sampled for all word tokens in a paper.

The different implementations reduces the process of writing a scientific paper to different series of probabilistic steps. They have different behaviors in the academic applications. In the remainder of this section, we will describe the three models in more detail.

6.2 ACT Model 1

In the first model (Figure 6(a)), the conference information is viewed as a stamp associated with each word in a paper. Intuition behind the first model is: coauthors of a paper determine topics written in this paper and each topic then generates the words and determines a proportion of the publication venue. The generative process can be summarized as follows:

1. For each topic z, draw ϕ_z and ψ_z respectively from Dirichlet priors β_z and μ_z;
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 $\psi_{z_{di}}$.

Following [26], we choose Gibbs sampling for inference. As for the hyperparameters α, β, and μ, for simplicity, we take a fixed value (i.e., $\alpha = 50/T$, $\beta = 0.01$, and $\mu = 0.1$). In the Gibbs sampling procedure, we first estimate the posterior distribution on just x and z and then use the results to infer θ, ϕ, and ψ. The posterior probability is calculated by the following:

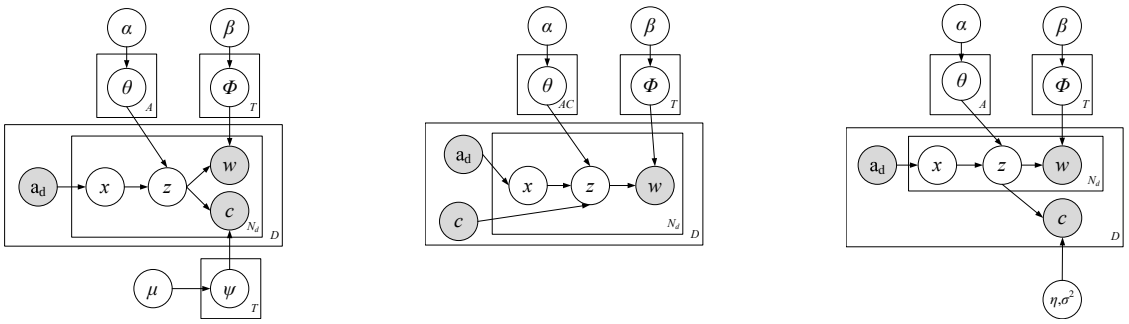
$$P(z_{di}, x_{di} | \mathbf{z}_{-di}, \mathbf{x}_{-di}, \mathbf{w}, \mathbf{c}, \alpha, \beta, \mu) \propto \frac{m^{-di}_{x_{di} z_{di}} + \alpha_{z_{di}}}{\sum_z (m^{-di}_{x_{di} z} + \alpha_z)} \frac{n^{-di}_{z_{di} w_{di}} + \beta_{w_{di}}}{\sum_{w_v} (n^{-di}_{z_{di} w_v} + \beta_{w_v})} \frac{n^{-d}_{z_{di} c_d} + \mu_{c_d}}{\sum_c (n^{-d}_{z_{di} c} + \mu_c)} \quad (5)$$

where the superscript $-di$ denotes a quantity, excluding the current instance (e.g., the di-th word token in the d-th paper).

After Gibbs sampling, the probability of a word given a topic ϕ, the probability of a conference given a topic ψ, and the probability of a topic given an author θ can be estimated as follows:

$$\acute{A}_{z w_{di}} = \frac{n_{z w_{di}} + \beta_{w_{di}}}{\sum_{w_v} (n_{z w_v} + \beta_{w_v})} \quad (6)$$

$$\tilde{A}_{z c_d} =$$



(a) ACT1 (b) ACT2 (c) ACT3

Figure 6: Graphical representation of the three Author-Conference-Topic (ACT) models.

In this model, the conference comes from a normal linear model. The covariates τ in this model are the frequencies of the topics in the document. The regression coefficients on these frequencies constitute η. The difference of parameterization from ACT1 is that the conference stamp is sampled from a normal linear distribution after topics were sampled for all word tokens in a paper.

For inference in ACT3, there is a slight difference from that in ACT1 and ACT2, as we also need to estimate the parameters η and σ^2. We use a Gibbs EM algorithm [2] for inference of this model.

In the E-step, for sampling the topic, the posterior probability is calculated by

$$P(z_{di}; x_{di} | \mathbf{z}_{-di}; \mathbf{x}_{-di}; c_d; \mathbf{w}; \text{®};$$

;xx-4.83TD[(di)]TJ/FF7riza98 Tf2507D[(z)]TJ/F9 5.98 TfTf 3.76 0 6.8D[(;)-177(z)]TJ/FF7riza98 Tf5056D[(z)]TJ/F9 5

uthor-C104A002cients on these frequencies

Topic #5 (Model 1) "Natural language processing"		Topic #10 (Model 1) "Semantic web"		Topic #16 (Model 1) "Machine learning"		Topic #19 (Model 1) "Support vector machines"		Topic #24 (Model 1) "Information extraction"	
language	0.034820	semantic	0.068226	learning	0.058056	support	0.082669	learning	0.065259
parsing	0.023766	web	0.048847	classification	0.018517	vector	0.071373	information	0.043527
natural	0.019029	ontology	0.043160	boosting	0.015881	machine	0.064076	extraction	0.033592
learning	0.015871	knowledge	0.041497	machine	0.017797	kernel	0.026897	web	0.019311
approach	0.012712	learning	0.013431	feature	0.013904	regression	0.020544	semantic	0.011860
grammars	0.012712	framework	0.012095	classifiers	0.013904	neural	0.016308	text	0.010618
processing	0.011923	approach	0.011427	margin	0.013245	classification	0.012072	rules	0.010618
text	0.011923	based	0.010758	selection	0.012586	networks	0.011366	relational	0.009376
Yuji Matsumoto	0.001389	Steffen Staab	0.005863	Robert E. Schapire	0.004033	Bernhard Scholkopf	0.003929	Raymond J. Mooney	0.010346
Eugene Charniak	0.001323	Enrico Motta	0.004365	Yoram Singer	0.003318	Johan A. K. Suykens	0.003536	Andrew McCallum	0.004074
Rens Bod	0.001323	York Sure	0.003713	Thomas G. Dietterich	0.002472	Vladimir Vapnik	0.002947	Craig A. Knoblock	0.003492
Brian Roark	0.001190	Nenad Stojanovic	0.001824	Bernhard Scholkopf	0.001496	OIvi L. Mangasarian	0.002947	Nicholas Kushmerick	0.002457
Suzanne Stevenson	0.001124	Alexander Maedche	0.001824	Alexander J. Smola	0.001301	Joos Vandewalle	0.002030	Ellen Riloff	0.002199
Anoop Sarkar	0.001058	Asuncion Gomez-Perez	0.001694	Ralf Schoknecht	0.001236	Nicola L. C. Talbot	0.001768	William W. Cohen	0.002134
ACL	0.253487	ISWC	0.125291	NIPS	0.289761	Neural Computation	0.096707	AAAI	0.295846
COLING	0.234435	EKAW	0.122379	JMLR	0.206583	NIPS	0.094388	IJCAI	0.192995
CL	0.118136	IEEE Intelligent Systems	0.071418	ICML	0.156389	ICANN	0.084338	ICML	0.060567
ANLP	0.060423	CoopIS/DOA/ODBASE	0.065594	COLT	0.096157	JMLR	0.083565	KDD	0.058551
CoRR	0.058674	K-CAP	0.054674	Neural Computation	0.023017	Neurocomputing	0.071197	JAIR	0.046451
COLING-ACL	0.036814	ESWS	0.023369	MLSS	0.011545	Machine Learning	0.067331	ECML	0.033006

Table 5: Five topics discovered by ACT1 on the Arnetminer data. Each topic is shown with the top 8 words and their corresponding probabilities. Top 6 authors and top 6 conferences are shown with each topic. The titles are our interpretation of the topics.

Table 6: Performance of six expertise search approaches (%).

Method	Object	P@5	P@10	P@20	R-pre	MAP
LM	Paper	40.0	38.6	37.1	10.0	46.4
	Author	65.7	44.3	25.0	58.8	73.4
	Conference	51.4	32.9	21.4	47.6	63.1
	Average	52.4	38.6	27.9	38.8	61.0
LDA	Paper	31.4	48.6	42.9	13.5	45.8
AT	Paper	42.9	48.6	42.9	13.1	49.3
	Author	82.9	45.7	25.7	73.5	78.1
	Average	62.9	47.1	34.3	43.3	63.7
ACT1	Paper	42.9	45.7	43.6	16.6	51.0
	Author	91.4	50.0	26.4	80.0	89.6
	Conference	62.9	41.4	23.6	60.7	72.3
	Average	**65.7**	**45.7**	**31.2**	**52.4**	**71.0**
ACT2	Paper	42.9	47.1	39.3	15.0	47.7
	Author	74.3	50.0	25.7	69.4	80.1
	Conference	54.3	41.4	22.1	54.2	63.9
	Average	57.1	46.2	29.1	46.2	63.9
ACT3	Paper	42.9	38.6	41.4	17.1	47.0
	Author	71.4	47.1	25.7	70.0	78.7
	Conference	57.1	38.6	23.6	58.3	65.7
	Average	57.1	41.4	30.2	48.5	63.8

Table 5 shows five topics discovered by ACT1.

Table 6 shows the experimental results of retrieving papers, authors, and conferences using our proposed methods and the baseline methods. We see that our proposed three methods outperform the baseline methods. LDA only models documents and thus can support only paper search; while AT supports paper search and author search. Both models underperform our proposed unified models. Our models benefit from the ability of modeling all kinds of information holistically, thus can capture the dependencies between the different types of information. We can also see that ACT1 achieves the best performance in all evaluation measures.

For comparison purposes, we also evaluate the results of two similar systems: Libra.msra.cn and Rexa.info. The average MAP obtained by Libra and Rexa on our data set are 48.3% and 45.0%. We see that our methods clearly outperform the two systems.

7.2 Applying ACT Models to Association Search

Association Search: Given a social network $G = (V; E)$ and an association query (a_i, a_j) (source person, target person), association search is to find and rank possible associations $\{\alpha_k(a_i, a_j)\}$ from a_i to a_j. Each association is denoted as a referral chain of persons.

There are two subtasks in association search: *finding* possible associations between two persons and *ranking* the associations. Given a large social network, to find all associations is an NP-hard problem. We instead focus on finding the 'shortest' associations. Hence, the problem becomes how to estimate the score of an association and one key issue is how to calculate the distance between persons. We use KL divergence to define the distance as:

$$KL(a_i; a_j) = \sum_{z=1}^{T} \mu_{a_i z} log \frac{\mu_{a_i z}}{\mu_{a_j z}} \quad (18)$$

We use the accumulated distance between persons on an association path as the score of the association. We call the association with the smallest score as the shortest association and our problem can be formalized as that of finding the near-shortest associations. Our approach consists of two stages:

1) Shortest association finding. It aims at finding shortest associations from all persons $a \in V \backslash a_j$ in the network to the target person a_j (the score of the shortest association from a_i to a_j is denoted as $L_{min} > 0$). We use a heap based Dijkstra algorithm to find the shortest associations.

2) Near-shortest associations finding. Based on the shortest association score L_{min} and a parameter γ, the algorithm uses a depth-first search to find associations whose scores are less than $(1 + \gamma)L_{min}$. We constrain the length of an association to be less than a pre-defined threshold. Finally, the obtained associations are ranked according to the scores.

Our approach can find the near-shortest associations for a query in less than 3 seconds on a commodity machine with a network of researchers. In the following, we list two associations ranked by our approach for the query ('Hang Li', 'Qiang Yang').

1. Hang Li -> Yong Yu -> Qiang Yang (score: 0.127)
2. Hang Li -> Bin Gao -> Wei-Ying Ma -> Qiang Yang (score: 0.274)

7.3 Other Applications

Our model can support many other applications, e.g., author interest finding and academic suggestion.

For example, Table 7 shows top 5 words and top 5 authors associated to two conferences found by ACT1. Table 8 shows top 5 words and top 5 conferences associated to two researchers found by ACT1. The results can be directly used to characterize the conference themes and researcher interests. They can be also used for prediction/suggestion tasks. For example, one can use the model to find the best matching reviewer for a paper submitted to a specific conference. Previously, such work is fulfilled by only keyword matching or topic-based retrieval such as [22], but not considering

Table 7: Top 5 representative words and top 5 authors associated to two conferences found by ACT1.

ACL		SIGIR	
parsing	0.030523	information	0.036946
semantic	0.018398	text	0.030265
learning	0.016851	classification	0.027953
statistical	0.014143	retrieval	0.025588
information	0.013620	web	0.021703
Christopher D. Manning	0.003984	Susan T. Dumais	0.002432
Dan I. Moldovan	0.003358	W. Bruce Croft	0.002190
Mark Johnson	0.002837	Norbert Fuhr	0.001643
Robert C. Moore	0.002055	Fabrizio Sebastiani	0.001279
Jason Eisner	0.001933	Laura A. Granka	0.001279

Table 8: Top 5 representative words and top 5 conferences associated to two researchers found by ACT1.

Raymond Mooney		Bruce Croft	
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
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

the conference. One can also use the model to suggest a venue to submit a paper based on its content and authors' interests. Or one can use it to suggest popular topics when authors prepare a paper for a conference.

8. CONCLUSION

In this paper, we describe the architecture and the main features of the ArnetMiner system. Specifically, we propose a unified tagging approach to researcher profiling. About a half million researcher profiles have been extracted into the system. The system has also integrated more than one million papers. We propose a probabilistic framework to deal with the name ambiguity problem in the integration. We further propose a unified topic model to simultaneously model the different types of information in the academic network. The modeling results have been applied to expertise search and association search. We conduct experiments for evaluating each of the proposed approaches. Experimental results indicate that the proposed methods can achieve a high performance.

There are many potential future directions of this work. It would be interesting to further investigate new extraction models for improving the accuracy of profile extraction. It would be also interesting to investigate how to determine the actual person number k for name disambiguation. Currently, the number is supplied manually, which is not practical for all author names. In addition, extending the topic model with link information (e.g., citation information) or time information is a promising direction.

9. ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China (90604025, 60703059), Chinese National Key Foundation Research and Development Plan (2007CB310803), and Chinese Young Faculty Research Funding (20070003093). It is also supported by IBM Innovation funding.

10. REFERENCES

[1] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27:187–203, 2005.

[2] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*.
86(netwode8.58 0 TD[(Moder2appremi27)-207ning)-500a-Yalei250(Doucet,)-25-342824(