Expertise Modeling for Matching Papers with Reviewers

David Mimno MIMNO@CS.UMASS.EDU
Andrew McCallum MCCALLUM@CS.UMASS.EDU
Department of Computer Science, University of Massachusetts, Amherst

Abstract

An essential part of an expert-finding task, such as matching reviewers to submitted papers, is the ability to model the expertise of a person based on documents. We evaluate several measures of the association between an author in an existing collection of research papers and a previously unseen document. We compare two language model based approaches with a novel topic model, Author-Persona-Topic (APT). In this model, each author can write under one or more "personas," which are represented as independent distributions over hidden topics. Examples of previous papers written by prospective reviewers are gathered from the Rexa database, which extracts and disambiguates author mentions from documents gathered from the web. We evaluate the models using a reviewer matching task based on human relevance judgments determining how well the expertise of proposed reviewers matches a submission. We find that the APT topic model outperforms the other models.

1. Introduction

Peer review is part of the foundation of the scientific method, but matching papers with reviewers can be a challenging and time-consuming process. The process is also a significant burden on the conference chair. There has been a recent trend towards bidding, which consumes additional reviewer time, as well as raising questions about the confidentiality of the submissions process. Matching papers with reviewers is a complicated task, with many sub-problems. Conference chairs must solve a complicated optimization problem involving constraints on the number of reviewers per paper and the number of papers per reviewer. One of the most important elements of the process, however, is modeling the expertise of a given reviewer with respect to the topical content of a given paper. This task is related to expert finding, an area which has received increased interest in recent years in the context of the TREC Enterprise Track. In addition, for or several years researchers in artificial intelligence have sought to automate, or at least streamline, the reviewer matching process.

In this paper, we evaluate several methods for measuring the affinity of a reviewer to a paper. These methods include language models with Dirichlet smoothing (Ponte & Croft, 1998; Zhai & Lafferty, 2001), the Author-Topic model (Rosen-Zvi et al., 2004), and a novel topic model, Author-Persona-Topic (APT).

We follow previous approaches in treating expert finding as an information retrieval task. The goal is to find relevant people rather than relevant documents, but we use the same basic tools. More specifically, we construct a model in which each potential reviewer has a distribution over words in the vocabulary, and then rank reviewers for a given paper based on the likelihood of the words in that paper under each reviewer's distribution. In this paper we evaluate several methods for constructing such models.

Statistical topic models represent documents as mixtures of topical components, which are distributions over the words in the corpus. The APT model is motivated by the observation that authors frequently write about several distinct subject area combinations. In the APT model, we not only learn the topical components, but also divide each author's papers into several "personas." Each persona clusters papers with similar topical combinations.

In order to discover expertise, it is necessary to consider how to represent expertise. It is rare that a person is an expert in all facets of a single topic. People usually describe their expertise as the combination of several topics, and often have experience in sev-

Presented at *North East Student Colloquium on Artificial Intelligence (NESCAI)*, 2007.

eral such intersections. For example, game theory and Bayesian networks or information retrieval and algorithms.

In order to learn which areas potential reviewers are experts in, it is necessary to have a training corpus of documents by or about those people. Previous work has been hampered by a lack of such training data. We take advantage of the Rexa database of research papers. This collection is built from research papers downloaded from the web. Rexa extracts information such as author names, titles and citations from PDF documents. Papers and authors are then coreferenced automatically.

Evaluating systems for reviewer matching is difficult. The actual assignments of reviewers to conference papers and the content of rejected papers is generally considered privileged information. Even if such data were available, it is not clear that such assignments necessarily represent the ideal matching, or simply a reasonable compromise to a difficult optimization problem. It is quite likely, for example, that reviewers not on a given panel may still be very relevant to the paper. As a result, we have collected human annotated relevance judgments for matchings between the reviewers and accepted papers for a recent Neural Information Processing Systems conference (NIPS 2006).

We measure the precision of each model after various numbers of reviewers have been retrieved. We find that a language model has the highest precision after five reviewers have been retrieved, but that the APT model with a relatively large number of topics has the highest precision at all other levels up to 30.

2. Related Work

The task of matching papers with reviewers has a long history. Dumais and Nielsen (1992) use Latent Semantic Indexing, trained on abstracts provided by prospective reviewers. Other approaches such as Benferhat and Lang (2001) take the affinity of reviewers to papers as given and concentrate on solving the optimization problem of constructing panels.

Rodriguez and Bollen (2006) present a system that propagates a particle swarm over a co-authorship network, starting with the authors cited by a submitted paper. The training corpus is the DBLP dataset, a manually maintained database of authors and research papers. The system is evaluated against self-described reviewer affinities from a recent conference (JCDL 2005).

Recent work by Hettich and Pazzani (2006) demonstrates the Revaide system for recommending panels of reviewers for NSF grant applications. Revaide uses a TF-IDF weighted vectors space model for measuring the association of reviewers with applications. The training corpus is the NSF database of "fundable" grant applications. Unfortunately, as with conferences, both the training corpus and the query document set for this study are confidential. Similarly, Basu et al. (1999) use web searches to find abstracts from papers written by reviewers, and then use a TF-IDF weighted vector space model to rank reviewers for a given submitted paper.

The inclusion of expert finding in the TREC Enterprise Track has resulted in a great deal of work on this area. One recent example is Balog et al. (2006), which presents two language models for expert finding.

The use of topic models for information retrieval tasks is described in Wei and Croft (2006). The authors find that interpolations between Dirichlet smoothed language models and topic models show significant improvements in retrieval performance above language models by themselves.

3. Modeling expertise

We evaluate several models of expertise. These can be divided into two main approaches: language models and topic models. In general, a language model based approach estimates the likelihood of a query given each document in the collection using a smoothed distribution derived from the words in that document. A topic model adds an additional level of representational power. Documents in the collection are represented as a mixture of topics, which are themselves mixtures of words.

Scientific publications frequently have more than one author. Rather than attempting to divide documents between authors, we simply replicate multi-author documents, once for each author. Although it is clear that the authors of a paper frequently focus on one aspect of that paper or another, we assume that all authors on a paper are at least substantially familiar with every aspect of the paper. In practice, replicating documents in this way has less effect in the reviewer matching application than in general expert finding, since we only consider authors who are in the list of reviewers. Therefore, documents will only be replicated if more than one author is also a reviewer.

3.1. Language models

In a language model, we represent each document as a multinomial over words. The maximum likelihood estimate of this multinomial is the number of times each word type appears in the document divided by the total number of tokens in the document. Since most words in the vocabulary do not appear in a given document, it is necessary to smooth the distribution. For all the models in this paper we use Dirichlet smoothing (Zhai & Lafferty, 2001). The likelihood of a query q consisting of some number of terms t for a document d under a language model with Dirichlet smoothing is

$$p(q|d) \quad = \quad \prod_{t \in q} \frac{N_d}{N_d + \mu} p(t|d) + \frac{\mu}{N_d + \mu} p(t) \qquad (1)$$

where N_d is the number of tokens in d, $p(t|d)$ is the maximum likelihood estimate described above, μ is a smoothing parameter, and $p(t)$ is the probability of the term in the entire corpus.

The first and simplest language model we evaluate is the single-document author model. In this model, for each author a we construct a document d_a, which is a concatenation of all documents in the corpus written by author a. The probability of a query given a reviewer r is therefore the probability of the query under Equation 2 given the author document d_r.

The second language model is the max-document author model. In this model we rank all documents for a given query using Equation 2, and then rank the reviewers in the order in which they first appear. We define D_r as the set of documents for which r appears as an author. The probability of a query given a reviewer under this model is thus

$$p(q|r) \quad = \quad \max_{d \in D_r} \prod_{t \in q} \frac{N_d}{N_d + \mu} p(t|d) + \frac{\mu}{N_d + \mu} p(t). (2)$$

The third language model is the document-sum author model. In this model, we calculate a maximum likelihood multinomial over all documents in the training corpus. For each term in the query, we calculate the probability of the term given the reviewer as the sum over all papers by that reviewer, smoothed by the probability of the term in the corpus as a whole. The probability of the query given a reviewer is therefore

$$p(q|r) \quad = \quad \prod_{t \in q} \left\{ (1-\lambda) \sum_{d \in D_r} p(t|d) \frac{1}{|D_r|} + \lambda p(t) \right\} (3)$$

This model is drawn from Petkova and Croft (2006), and is similar to Model 1 from Balog et al. (2006). We follow Petkova and Croft in setting $\lambda = 0.1$.

The three language models approach relevance in different ways. In the single-document model, most of a reviewer's work must be similar to a given paper in order for that reviewer to be ranked highly, but no particular document needs to exactly match the submission. In contrast, in the max-document model, a reviewer must have at least one document that very closely matches the word distribution of the paper. The document-sum model is in a way a compromise between these two: a single relevant document will not be "washed out" by a large body of non-relevant work, but the author of one highly relevant document (among many) will not necessarily be ranked higher than the author of many slightly less relevant documents.

The smoothing parameters for the language models are chosen to be the average length of the documents in the training corpus for each model. Since the documents in the single-document author model are generally much larger than the documents in the max-document author model, the smoothing parameter for this model tends to be much larger, approximately 2000 vs. approximately 50.

Other published work such as Hettich and Pazzani (2006) uses TF-IDF weighting in a vector space model. We do not evaluate a vector space model here, but it has been shown both that language model based information retrieval systems outperform TF-IDF based systems (Ponte & Croft, 1998) and that Dirichlet smoothing in language models implies the effect of both TF-IDF weighting and document length normalization (Zhai & Lafferty, 2001).

3.2. Topic models

A statistical topic model represents a topic as a distribution over words, which is drawn from a Dirichlet prior. In a simple topic model such as Latent Dirichlet Allocation (Blei et al., 2003), each document has a distribution over topics. Words are generated by selecting a topic from the document's topic distribution, and then selecting a word from that topic's distribution over the vocabulary. Although directly optimizing the topic-word and document-topic distributions is intractable, these models can be trained efficiently using Gibbs sampling. Topic models produce interpretable, semantically coherent topics, which can be examined by listing the most probable words for each topic.

Statistical topic models have been previously used to describe the topical distributions of authors, for example the Author-Topic model by Rosen-Zvi et al. (2004) and the Author-Recipient-Topic model by McCallum et al. (2005). In the Author-Topic (AT) model, each author has a distribution over topics, unlike the simple topic model where each document has its own topic distribution. Under the AT generative model, a document has some number of authors, whose identity is observed. Each word is generated by selecting one of those authors, sampling a topic from that author's topic distribution, and then sampling a word from that topic's distribution over the vocabulary. Note that one of the goals of the AT model is to learn which author is responsible for a given word in a document. We avoid this question entirely by replicating documents that have more than one reviewer as an author. This decision is based on our goals for the model: we want to discover a broader notion of which combinations of topics a given reviewer is competent to review, rather than to judge the relative strengths of coauthors in a particular paper.

The topic models are trained by Gibbs sampling. In all cases we average over the results of 10 Gibbs sampling chains.

3.2.1. Author-Topic model

For this paper, we evaluate two topic models. The first is a simplified version of the AT model. All training documents in the corpus are constrained to have a single author, so the variables representing which author is responsible for a given word are meaningless. The resulting model can be thought of as a simple topic model run on the concatenated documents described earlier in the language model section for the single-document author model.

The generative model for the single-author AT model can be described by the following Bayesian hierarchical model. The model includes two Dirichlet hyperparameters, α and β, which are the size of the set of topics and the vocabulary of the corpus, respectively.

1. For each topic t sample a multinomial over words ϕ_t from β.
2. For each author a sample a multinomial over topics θ_a from α.
3. For each document d with author a,
   (a) For each token i
      i. Sample a topic z_i from θ_a.
      ii. Sample a word w_i from ϕ_{z_i}.

The probability of the words and topic assignments of the entire corpus is then

$$p(\mathbf{w}, \mathbf{z}, \phi, \theta | \mathbf{a}, \alpha, \beta) = \prod_d \prod_i p(w_{di} | z_{di}, \phi_{z_{di}}) p(z_{di} | \theta_{a_d}) \times \prod_t p(\phi_t | \beta) \prod_a p(\theta_a | \alpha). \tag{4}$$

Rearranging the terms to group the words and topics drawn from each multinomial and integrating over the multinomial parameters ϕ and θ, we are left with two products over Dirichlet-multinomial distributions. These depend on the hyperparameters and certain statistics of the corpus: N_t^v, the number of words of type v in topic t, N_a^t, the number of words of topic t in documents by author a, N_t, the total number of words in topic t, and N_a, the total number of words written by author a.

$$p(\mathbf{w}, \mathbf{z}, \phi, \theta | \mathbf{a}, \alpha, \beta) = \prod_a \frac{\Gamma \sum_t \alpha_t}{\prod_t \Gamma \alpha_t} \frac{\prod_t \Gamma(\alpha_t + N_a^t)}{\Gamma \sum_t (\alpha_t + N_a^t)} \times \prod_t \frac{\Gamma \sum_v \beta_v}{\prod_v \Gamma \beta_v} \frac{\prod_v \Gamma(\beta_v + N_t^v)}{\Gamma \sum_v (\beta_v + N_t^v)} \tag{5}$$

The predictive distribution for Gibbs sampling can be derived as the probability of adding a word of type v written by author a to a topic t. This is

$$p(t|v, a) \propto \frac{\alpha_t + N_a^t}{\sum_t (\alpha_t + N_a^t)} \frac{\beta_v + N_t^v}{\sum_v (\beta_v + N_t^v)}. \tag{6}$$

The term $\sum_t (\alpha_t + N_a^t)$ is constant with respect to t, but is included here for clarity. We train the topic model for 1000 iterations of Gibbs sampling.

Once we have a trained topic model, the next step is to derive the likelihood of a query given the model. Here we follow Wei and Croft (2006). We estimate the multinomial parameters using expressions similar to the predictive distribution above.

$$p(v|\hat{\phi}_t) = \frac{\beta_v + N_t^v}{\sum (\beta_v + N_t^v)} \tag{7}$$

$$p(t|\hat{\theta}_a) = \frac{\alpha_t + N_a^t}{\sum_t (\alpha_t + N_a^t)} \tag{8}$$

Finally, we represent the probability of a term given an author as a weighted sum over all topics of the probability of the word given the topic. The probability of a query (here we use v to represent query terms to avoid confusion) is therefore the product of the probabilities of the terms:

$$p(q|a) = \prod_{v \in q} \sum_t p(v|\hat{\phi}_t) p(t|\hat{\theta}_a). \tag{9}$$

3.2.2. Author-Persona-Topic model

In addition to the single-author AT model, we present a novel topic model, the Author-Persona-Topic (APT) model. The difference between APT and AT is that rather than grouping all papers by a given author under a single topic distribution, we allow each author's documents to be divided into one or more clusters, each with its own separate topic distribution. These clusters represent different "personas" under which a single author writes.

An important question is how many potential personas each author should have. In this work we set the number of personas for author a to $\lceil |D_a|/20 \rceil$. Thus each author has at least one persona, and one additional persona for every twenty papers. We experimented with setting the number of personas proportional to the log of the number of papers and with allowing the model to choose a number of personas using a non-parametric prior. Neither method was as effective as the linear number of personas; results for those models are not reported here.

The generative model for APT is as follows. The hyperparameters are the same as with the AT model, except for the addition of a hyperparameter for the distribution over personas for each author. Since authors have varying numbers of personas, we cannot draw all distributions over personas from the same Dirichlet parameter for every author. Therefore we define a separate Dirichlet parameter γ_a for every author, all set to a symmetric distribution with $\gamma_{a_g} = 10$.

1. For each topic t sample a multinomial over words ϕ_t from β.
2. For each author
   (a) Sample a multinomial over personas η_a from γ_a.
   (b) For each persona g in a sample a multinomial over topics θ_g from α.
3. For each document d with author a_d,
   (a) Sample a persona g_d from η_{a_d}
   (b) For each token i
      i. Sample a topic z_i from θ_{g_d}.
      ii. Sample a word w_i from ϕ_{z_i}.

The probability of the entire corpus is therefore

$$p(\mathbf{w}, \mathbf{z}, \mathbf{g}, \eta, \phi, \theta | \mathbf{a}, \alpha, \beta, \gamma) = \prod_d \left[p(g_d|\eta_{a_d}) \prod_i p(w_{di}|z_{di}, \phi_{z_{di}}) p(z_{di}|\theta_{g_d}) \right] \times \prod_t p(\phi_t|\beta) \prod_g p(\theta_g|\alpha) \prod_a p(\eta_a|\gamma_a) \quad (10)$$

As with the AT model, we use Gibbs sampling to draw samples from this distribution conditioned on the words and authorships in the corpus. For each document, we sample the topic assignment for each word and then the persona assignment for the document. The predictive distribution for the each word's topic assignment is the same as Equation 6, substituting g_d for a. Sampling the persona assignment of an entire document is more difficult, since all of the word-topic assignments depend on the persona. In order to sample a new persona, we remove the current setting of g_d from N_a^g (the number of documents by author a assigned to persona g) and remove all topic counts for the document from $N_{g_d}^t$. We represent the number of tokens assigned to topic t in documents other than d that are assigned to persona g_d as $N_{g_d \setminus d}^t$. The predictive distribution for a persona given all the word-topic assignments $\mathbf{z}_d$ is

$$p(g_d|\mathbf{z}, a, \gamma_a) \propto \frac{\gamma_{a_g} + N_a^g}{\sum_{a_g} (\gamma_{a_g} + N_a^g)} \times \frac{\Gamma \sum_t (\alpha_t + N_{g_d \setminus d}^t)}{\prod_t \Gamma(\alpha_t + N_{g_d \setminus d}^t)} \frac{\prod_t \Gamma(\alpha_t + N_{g_d}^t)}{\Gamma \sum_t (\alpha_t + N_{g_d}^t)} \quad (11)$$

This represents the probability of picking persona g_d given the number of documents assigned to that persona and the total number of documents for author a, as well as adding some number of words to each topic, beyond the number of words in that topic from other documents in the persona.

4. Evaluation

It is difficult to evaluate the quality of paper/reviewer relevance rankings due to the scarcity of data that can be examined publicly. As a result, we approximate the task of assigning reviewers to submitted papers by gathering experts' relevance judgments from humans for rankings of reviewers and accepted papers for the NIPS 2006 conference. We in fact use the reviewer list from NIPS 2005, as we were unable to find the list of reviewers for NIPS 2006, but we do not believe that the difference is significant.

We evaluate our algorithms on the resulting list of 148 papers and 364 reviewers. It would be very difficult and time-consuming to gather relevance judgments for every combination of reviewers and papers, most of which will not be relevant. As a result, we use pooled relevance judgments (Buckley & Voorhees, 2004). In this method, we ask each model to rank the reviewers

Table 1. Sample topics from the APT model with 200 topics on a corpus of about 500,000 words. The documents consist of titles and abstracts from papers written by NIPS reviewers. The column on the left is the total number of words in each topic, while the column on the right is a listing of the most probable words for each topic.

N_t	Most probable words
23574	performance data results training set
42871	problem results show time problems
28737	data model algorithm method methods
7604	models model hidden markov mixture
9031	vector support machines kernel svm
1844	fields extraction random conditional sequence
1961	information method bottleneck memory classification
3858	models conditional discriminative maximum entropy
8806	speech recognition acoustic automatic features
3143	carlo monte sampling chain markov
1642	bias variance error cross estimator
2012	reinforcement control agent rl search
4092	language word words english statistical
2679	expression gene data genes binding
4617	software development computer design research
1131	objects nodes world semantic show
769	geometric patterns pattern dimensional noise

for each paper. We then take the top five reviewers from each ranked list and merge them, removing duplicates. This pool of reviewers is then presented to human annotators. Since we remove duplicates, pools for papers that the models showed substantial agreement are smaller than pools for papers in which the models disagreed.

We asked several prominent researchers from the NIPS community to mark the relevance of the proposed reviewers. Each reviewer was encouraged to select papers from the conference proceedings that were particularly related to his or her research. We collected a total of 650 reviewer/paper relevance judgments from nine annotators.

We used a four-level relevance scheme, as follows: Very Relevant (3), Relevant (2), Slightly Relevant (1) and Irrelevant (0).

We evaluate the results using `trec_eval`.[1] The evaluation algorithms implemented in this package are defined only for binary relevance judgments. We therefore evaluate each algorithm under two relevance cutoffs, such that either 2 or 3 are considered relevant or only 3 is considered relevant. If there are disagreements between annotators we default to the lower ranking.

[1] ftp://ftp.cs.cornell.edu/pub/smart

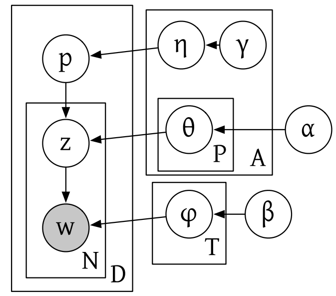

Figure 1. A graphical model representation of the Author-Persona-Topic model. Each author has some number of personas, each represented by a multinomial θ. To generate a document, an author chooses a persona p, distributed according to η, and then selects topics from θ_p.

Examples of topics from a model trained with 200 topics are shown in Table 1. The model is able to identify and separate common methodological words ("performance, data, results" and "data, model, algorithm") while also identifying clusters of words related to specific machine learning algorithms: there are topics for hidden markov models, support vector machines, information bottleneck and conditional random fields.

The personas discovered by the APT model are also coherent and interpretable. Examples of personas for two Computer Science researchers are shown in Table 4 (David Karger) and Table 5 (Daphne Koller). We also list in the captions of those tables subject terms used by the researchers themselves on their publications web pages. In both cases the APT model has essentially rediscovered the organization that the researchers themselves chose for their own papers. For example, Karger's largest persona includes topics related to algorithms and graphs; he lists "Cuts and Flows" as a major area of research. Other personas include topics related to peer-to-peer networking ("Applications of Theory") and web search ("Information Retrieval"). Koller also identifies areas discovered by the APT model, such as "Computational Biology" and "Computational Game Theory."

Results for precision at various numbers of reviewers returned for both relevance cutoffs are shown in Tables 2 and 3. There is a marked difference in performance between the topic models with 200 topics and with 75 topics. In general the models with more fine grained topics do better.

In most cases, the APT topic model with 200 topics has the highest precision. At the 5-reviewers level, the single-document author language model performs best. This is not particularly surprising: if all of an author's work matches closely with a query document, it is very likely that that person is a good reviewer for that paper. In other cases, the contextual smoothing provided by the topic models is better at finding relevant reviewers.

Table 2. Precision at relevance cutoff ≥ 2 after retrieving n reviewers.

Model	5	10	15	20	30
APT 200	0.4118	**0.2971**	**0.2255**	**0.1824**	**0.1294**
AT 200	0.3882	0.2765	0.2176	0.1794	0.1265
max-doc	0.3471	0.2500	0.1980	0.1588	0.1147
single-doc	**0.4471**	0.2735	0.1980	0.1529	0.1059
doc-sum	0.3412	0.2500	0.1882	0.1529	0.1118
APT 75	0.3059	0.2588	0.1961	0.1618	0.1176
AT 75	0.3529	0.2588	0.2020	0.1632	0.1275

Table 3. Precision at relevance cutoff 3 after retrieving n reviewers.

Model	5	10	15	20	30
APT 200	0.2059	**0.1412**	**0.1059**	**0.0824**	**0.0569**
AT 200	0.1882	0.1324	0.0980	0.0809	0.0549
max-doc	0.1765	0.1176	0.0961	0.0721	0.0510
single-doc	**0.2235**	0.1206	0.0902	0.0676	0.0451
doc-sum	0.1529	0.1206	0.0843	0.0676	0.0480
APT 75	0.1412	0.1147	0.0902	0.0721	0.0520
AT 75	0.1529	0.1147	0.0941	0.0765	0.0549

5. Discussion and Future Work

We have shown that statistical topic models can be an effective tool in expert retrieval in the context of matching papers with reviewers. Language models with Dirichlet smoothing also perform well, especially in finding the most relevant reviewers. We find that topic models are sensitive to the number of topics, with more topics providing a substantial performance boost. There are many areas for future work, such as taking advantage of citations and co-authorship data and building language models based on the partition of an author's papers provided by the APT model.

Ultimately, measuring the expertise of a person given a paper is only a part of a system for matching reviewers to papers. As probabilistic models, the methods described in this paper could fit easily into a larger likelihood function that takes into account the number of reviewers per paper and the number of papers per reviewer. Finding a good matching for the conference as a whole would then be a matter of sampling matchings with high probability from that model. The highly constrained nature of such optimization problems suggests that the additional accuracy of the topic modeling approaches at the 10 reviewer level could be valuable.

6. Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and in part by NSF grant # CNS-0551597, in part by NSF Nano # DMI-0531171, and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

Balog, K., Azzopardi, L., & de Rijke, M. (2006). Formal models for expert finding in enterprise corpora. *SIGIR*.

Basu, C., Hirsh, H., Cohen, W., & Nevill-Manning, C. (1999). Recommending papers by mining the web. *IJCAI*.

Benferhat, S., & Lang, J. (2001). Conference paper assignment. *Int. J. Intell. Syst*, *16*, 1183.

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Buckley, C., & Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. *SIGIR*.

Dumais, S. T., & Nielsen, J. (1992). Automating the assignment of submitted manuscripts to reviewers. *SIGIR* (p. 233).

Hettich, S., & Pazzani, M. J. (2006). Mining for proposal reviewers: lessons learned at the national science foundation. *KDD*.

McCallum, A., Corrada-Emmanuel, A., & Wang, X. (2005). Topic and role discovery in social networks. *IJCAI*.

Petkova, D., & Croft, W. B. (2006). Hierarchical language models for expert finding in enterprise corpora. *International Conference on Tools with Artificial Intelligence*.

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. *SIGIR*.

Rodriguez, M. A., & Bollen, J. (2006). *An algorithm to determine peer-reviews* (Technical Report). Los Alamos National Laboratory.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. *UAI*.

Table 4. Author-Persona-Topic distributions for David Karger, sorted by the number of papers per persona. For comparison, the categories Karger lists on his publications web page include "Information Retrieval," "Applications of Theory" (which includes the Chord peer-to-peer lookup protocol), "Cuts and Flows" and "Graph Coloring." The number on the left is the number of words assigned to each topic within the persona.

N_g^t	Persona 1 topic words [64 papers] **Cuts and Flows**
1724	time minimum randomized problem cut algorithm network approximation
359	algorithm algorithms problem problems results efficient function techniques
303	show time function optimal number case results constant
238	graph graphs edges directed edge general nodes link
222	show set data method information number simple linear
N_g^t	Persona 2 topic words [35 papers] **Applications of Theory**
1062	peer users user web semantic chord distributed rdf
215	information network knowledge content wide people sharing file
159	large describe natural previous results small type result
155	dynamic control design fast high simulation complex tasks
143	system systems information performance results task data techniques
N_g^t	Persona 3 topic words [15 papers] **Information Retrieval**
200	text documents web search document topic retrieval extraction
148	algorithm algorithms problem problems results efficient function techniques
80	show set data method information number simple linear
59	dynamic control design fast high simulation complex tasks
47	classification training classifier classifiers error performance bayes class

Table 5. Author-Persona-Topic distributions for Daphne Koller, sorted by the number of papers per persona. Koller annotates papers on her publications web page with topical labels. These include "Bayesian Networks," "Computational Biology," "Computational Game Theory," "Learning Graphical Models," "Natural Language," "Text and Web" and "Theoretical Computer Science"

N_g^t	Persona 1 topic words [48 papers] **Bayesian Networks**
980	probabilistic representation reasoning relational language world objects networks
224	show set data method information number simple linear
145	bayesian inference networks models graphical variables approximate probabilistic
N_g^t	Persona 2 topic words [29 papers] **RL and Dynamic Bayesian Networks**
299	algorithm algorithms problem problems results efficient function techniques
285	learning state reinforcement decision policy markov time actions
268	bayesian inference networks models graphical variables approximate probabilistic
N_g^t	Persona 3 topic words [20 papers] **Computational Game Theory**
263	games game equilibria nash agent strategies equilibrium strategy
137	algorithm algorithms problem problems results efficient function techniques
136	show set data method information number simple linear
N_g^t	Persona 4 topic words [18 papers] **Computational Biology**
159	belief bayesian structure networks variables gene expression search
109	gene protein expression dna genes binding sequence motifs
78	data sets real classification representation world classes datasets
N_g^t	Persona 5 topic words [9 papers] **Text and Web**
71	conditional fields models random discriminative structured sequence label
45	model models data modeling probabilistic parameters structure analysis
26	text documents web search document topic retrieval extraction

Wei, X., & Croft, B. (2006). LDA-based document models for ad-hoc retrieval. *SIGIR 2006*.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. *SIGIR*.