Exploring Social Annotations for Information Retrieval

Ding Zhou*
Facebook Inc.
156 University Avenue
Palo Alto, CA, 94301

Jiang Bian
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

Shuyi Zheng
Computer Science & Engineering
Pennsylvania State University
University Park, PA 16802

Hongyuan Zha
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

C. Lee Giles
Information Sciences and Technology
Computer Science & Engineering
Pennsylvania State University
University Park, PA 16802

ABSTRACT

Social annotation has gained increasing popularity in many Web-based applications, leading to an emerging research area in text analysis and information retrieval. This paper is concerned with developing probabilistic models and computational algorithms for social annotations. We propose a unified framework to combine the modeling of social annotations with the language modeling-based methods for information retrieval. The proposed approach consists of two steps: (1) discovering topics in the contents and annotations of documents while categorizing the users by domains; and (2) enhancing document and query language models by incorporating user domain interests as well as topical background models. Differences in user domain expertise are also considered when combining the discovered user domain interests. In particular, we propose a new general generative model for social annotations, which is then simplified to a computationally tractable hierarchical Bayesian network. Then we apply smoothing techniques in a risk minimization framework to incorporate the topical information to language models. Experiments are carried out on a real-world annotation data set sampled from del.icio.us. Our results demonstrate significant improvements over the alternative approaches without consideration of topical information, social annotations, user expertise, or simple incorporation of topic analysis.

1. INTRODUCTION

The goal of the semantic Web [1] is to make the Web resources understandable to both humans and machines. This has motivated a stream of new Web applications including Web blogs [10], social annotations (a.k.a social bookmarking) [4, 3, 20], and Web social networks [22]. Research in Web blogs and social networks has been especially focused on discovering the latent communities [10, 22], detecting topics from temporal text streams [14], and the retrieval of such highly dynamic information. In this paper, we focus on the social annotations [1] in large part motivated by their increasing availability across many Web-based applications.

Social annotation is a form of *folksonomy*, which refers to Internet-based methods for collaboratively generating open-ended text labels that categorize content such as Web pages, online photographs, and Web links. Many popular Web services rely on folksonomies including *delicious* (del.icio.us) and *flickr* (flickr.com). Despite the rising popularity of those Web services, research on in folksonomies is still at an early stage. Much of the work has been focused on the study of the data properties, the analysis of usage patterns of tagging systems [4], and the discovery of hidden semantics in tags [20]. The objective of this paper, however, is to leverage the efforts and expertise of users embodied in social annotations for improving user experience in *information retrieval* (IR). We advance previous work by combining topic analysis with language modeling methods used in contemporary IR [6].

Incorporating social annotations with document content is a natural idea, especially for IR applications. Consider the IR methods based on language modeling, for example [15, 12], we may simply treat the terms in annotation tags the same as those in document content, consider them as additional terms of the documents, and then follow the existing IR approaches. The pitfalls here, however, come in several forms. First, a tag term is generated differently than a document content term. A tag, upon its generation by a user, represents an abstract of the document from *single* perspective of a *single* user. Second, the differences in domain expertise of users should be taken into consideration when incorporating user tags. Some users in certain domains might be more trustworthy than others. Some users for various reasons may give incorrect tags. Although it remains an open problem to discover domain expertise of users, such peer differences are believed to be important [7] for effective societal information retrieval. Finally,the im-

*This work was done at The Pennsylvania State University.

WWW 2008, Beijing, China
ACM 978-1-60558-085-2/08/04.

[1] By a social annotation, we mean the annotation tags associated with the document. Each tag is generated by a user (or shared by several users) that can include several terms.

provement for IR will be limited without considering the semantics of the tag terms. Usually the number of tag terms is much smaller than the number of terms in a document being tagged. Therefore using the tag terms in the same way as the document terms are used will lead to the same problems observed in traditional language modeling-based IR, such as the lack of smoothness of results and the sparsity of observations.

In this paper, we develop a framework that combines the modeling of social annotations with the expansion of traditional language modeling-based IR using user domain expertise. First, we seek to discover topics in the content and annotations of documents and categorize the users by domains. We propose a probabilistic generative model for the generation of document content as well as the associated tags. Second, we follow an IR framework based on risk minimization proposed earlier [12]. The framework is based on Bayesian decision theory focusing on improving language models for queries and documents. We then study several ways for expanding the language models where the user domain interests and expertise and the background collection language models are incorporated. In particular, we apply linear smoothing between the original term-level language models and the new topic-level language models. The newly proposed framework benefits from the consideration of the differences between document content terms and tag terms in the modeling process. User domain expertise can be readily included in the retrieval framework by the proposed ways of language model expansion. The smoothing of the original term-level language model with the topic-level language models addresses the issues raised by the sparsity of observations.

The main contributions of this paper include (1) a general and a simplified probabilistic generative model for the generation of document content as well as the associated social annotations; (2) a new way for categorizing users by domains based on social annotations. The user domain expertise, evaluated by activity frequency, are used to weigh user interests; (3) the study of several ways for combining term-level language models with those topic-level models obtained from topics in documents and users.

The rest of this paper is organized as follows: Sec. 2 introduces the related work on topic analysis and language modeling. Sec. 3 proposes the new probabilistic generative models for the social annotations, including a brief discussion on choosing the correct topic number; In Sec. 5, we review the risk minimization framework for information retrieval as a Bayesian decision process. Sec. 6 explores several methods for incorporating the discovered domain interests to language modeling-based IR. Experimental results are presented in two sections, Sec. 4 and Sec. 7, respectively for topic analysis and IR quality. We conclude the paper and discuss future work in Sec. 8.

2. RELATED WORK

We review two lines of work which are closely related to the approach we will propose; the document content characterization, and information retrieval based on language modeling.

2.1 Topic Analysis using Generative Models

Related work on document content characterization [2, 17, 13, 18, 22] introduce a set of probabilistic models to simulate the generation of a document. Several factors in producing a document, either observable (e.g. author [17, 18]) or latent (e.g. topic [2, 13], community [22]), are modeled as variables in the generative Bayesian network and have been shown to work well for document content characterization. The Latent Dirichlet Allocation (LDA) model [2] is based upon the idea that the probability distribution over words in a document can be expressed as a mixture of topics, where each topics is a probability distribution over words. Along the line of LDA, the Author-Word model proposed in [13] considers the interests of single authors as the origin of a word. Influential following work named Author-Topic model combines the Topic-Word and Author-Word models, such that it regards the generation of a document as affected by both factors in a hierarchical manner [17, 18]. A recent work on social network analysis extends the previous model with an additional layer that captures the community influence in the setting of information society. The model proposed in this paper is related but different from the Author-Topic model proposed before [17]. Here the users or the sources of the tags and documents are observed instead of being latent in the generation process.

2.2 Information Retrieval based on Language Modeling

This work also overlaps with the research on information retrieval (IR) using probabilistic language modeling. Language modeling is a recent approach to IR which is considered as an alternative to traditional vector space models and other probabilistic models. This approach was initially proposed by Ponte and Croft [15]. The basic idea is to estimate the probability of generating the query from the candidate documents, each of which is modeled as a language model. The research line in IR using language models is later supported [12] by a framework based on Bayesian decision theory, which transforms the focus into improving the language models. A common way for improving language model is smoothing, which seeks to fight against the challenge of estimating an accurate language model from the insufficient data available. A relative complete study of the smoothing methods for statistical language modeling is given in [21]. Usually the document language model is smoothed with the background collection model, a pre-built model believed to be smoother and contain more words. This paper employs the linear interpolation [9] of the original language model with the reference models discovered before. This way, the social expertise of the users are imported to the language modeling and will further improve the quality of information retrieval. In addition to the above traditional work, a recent work [20] presents a preliminary study on clustering annotations based on EM for semantic Web and search. The probability of seeing certain words for a URL is estimated, which is then used for retrieval. However, the URL content and differences in users are not considered in that work.

3. MODELING SOCIAL ANNOTATIONS

We propose a probabilistic generative model for social annotations. The model specifies the generation process of the terms in document content as well as the associated user tags. The motivation for modeling the social annotations with document content is to obtain a simultaneous topical

analysis of the terms, documents, as well as the users. As we will discuss later, the topical analysis of terms (or the clustering of them by topics) essentially provides the basis for expanding query and document language models. In addition, the topical analysis of users, which categorizes the users by domains, enables the input of domain expertise of users in addition to the tags generated by them. This section starts with the introduction to modeling the user tag generation, as effected by document content. Then we simplify the general generative model for tags to a structure which is tractable and easier to estimate. Finally, we present the training method and a discussion on selecting the number of topics using the perplexity measure.

3.1 Generative Models for Annotations

We start by modeling the generation of words in documents and annotations. Intuitively, the content of documents and annotations are generated by two similar but correlated approaches. We illustrate our understanding of the generation process in plate notation in Fig. 1. On the document side (left-hand side), for an arbitrary word ω in document d, a topic z is first drawn, then conditioned on this topic, ω is drawn; Repeating this process for N_d times, which is the number of words in d, d is generated. The whole collection repeats the same process for D times [2]; On the annotation side (right-hand side), each word in the annotation is generated similarly. First, an observed user a decides to make annotation on a particular document, then the user picks a topic z to describe the d, followed by the generation of ω. The generation of z by user, however, depends not only on the user but also the topic of d. Note the dependency of user topics on document topics can be seen as a mapping between two conceptions. Generally speaking, there are different number of topics on both sides, T_d and T_a. The two topic sets can be different but are usually very similar.

Inspired by related work on topic analysis [2, 17, 18], we make assumptions about the probability structures of the generative model in Fig. 1. First, we assume all the conditional probabilities follow multinomial distribution. For example, each topic is a multinomial distribution over words where for the conditional probability of each word is fixed. Second, we assume that the prior distribution for topics and words follow Dirichlet (θ_d,ϕ_d for documents and θ_a,ϕ_a for annotations), which are conjugate priors for multinomial, respectively parameterized by α_d,β_d and α_a, β_a.

The generative model, illustrated in Fig. 1, is not quite tractable in practice. The probability distributions we would have to estimate include: (1) $D + A$ multinomial distributions for documents over topics; (2) $T_d + T_a$ multinomial distributions for topics over words; (3) $T_d \times T_a$ conditional probabilities to capture the correlation of the topics in documents and the topics in annotations. In addition, there are many parameters that adds difficulty in tuning in practice (α_d, β_d, α_a, β_a, T_d, and T_a). Therefore, in the next section, we will simplify this general annotation model with some relaxations in assumptions, arriving at a tractable model with easy training algorithms available.

[2]Note the document side of the general annotation model is essentially the LDA model proposed in [2]. But the right side takes into consideration the generation of annotations as dependent on the document content generation.

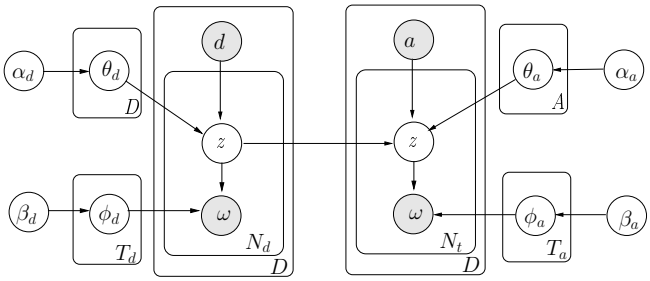

Figure 1: The general generative model for content of documents and annotations in plate notation. T_d (T_a) is the number of topics in documents (annotations); N_d (N_t) is the number of content words (or tag words) in document d; A and D are the number of users and documents; θ_d, θ_a ϕ_d, and ϕ_a are Dirichlet priors parameterized respectively by, α_d, α_a, β_d, and β_a. Dark circles denote the observed variables and the blank circles denote the hidden ones. Rectangles denote the repetition of models with the same structure but different parameters, where the lower-right symbol indicates the number of repetitions.

3.2 A Simplified Annotation Model

In this section, we simplify the general annotation model given before. In order to reduce the general model to a one tractable with fewer parameters, we make several compromises in assumptions. First, we assume the topics in documents and annotations are the same. This assumes that the taggers conceptually agree with the original document authors without variation of information in their understanding. Second, we assume that documents and users have the same structure of prior distributions which are only parameterized differently. Although arguably the users and documents might have different types of distributions over topics, we make the assumption here for the sake of simplicity.

The assumptions before lead to a simplified generative model for annotations. As illustrated in Fig. 2, we have a single topic-word distribution ϕ with parameter β; a single source-topic distribution with extended dimension (here the source can be a document or a tagger). Now we have much fewer distributions to estimate, making the modeling more tractable in practice.

Let us name the the model in Fig. 2 as the user-content-annotation (UCA) model. The UCA model describes the generation of words in document content and in the tags in similar but different processes. For document content, each observed term ω in document d is generated from the source x (each document d maps one-to-one to a source x). Then from the conditional probability distribution on x, a topic z is drawn. Given the topic z, ω is finally generated from the conditional probability distribution on the topic z. For document tags, similarly, each observed tag word ω for document d is generated by user x. Specific to this user, there is a conditional probability distribution of topics, from which a topic z is then chosen. This hidden variable of topic again finally generates ω in the tag.

According to the model structure, the conditional joint probability of θ, ϕ, x, z, ω given the parameters α, β is:

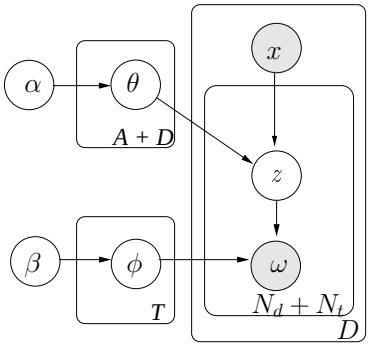

Figure 2: The User-Content-Annotation (UCA) Model in plate notation. T, A, and D are the number of topics, users, and documents. N_d and N_t denote the number of terms in the document and the number of terms in the tag. ϕ is the topic-word distribution with parameter β; θ is the source-topic distribution with parameter α.

$$P(\theta, \phi, x, z, w|\alpha, \beta) = \quad (1)$$
$$P(w|z, \phi)P(\phi|\beta)P(z|x, \theta)P(\theta|\alpha)P(x); \quad (2)$$

For inferences of words, we can calculate the conditional probability given a word as:

$$P(\theta, \phi, x, z, |\omega, \alpha, \beta) = \frac{P(\theta, \phi, x, z, \omega|\alpha, \beta)}{\sum_x \sum_z P(\theta, \phi, x, z, \omega|\alpha, \beta)}. \quad (3)$$

Again, similar to related work, we make assumptions regarding the probability structures. We assume the prior distribution of topics and terms follow Dirichlet distributions parameterized respectively by α and β. Let T be the number of topics (input as a parameter); A is the number of users; D is the number of documents; N_d and N_t respectively denote the number of terms in the document and the number of terms in the tag. Each topic is a probabilistic multinomial distribution over terms, denoted by ϕ; Each user (or source) is a probabilistic multinomial distribution over topics, denoted by θ. As illustrated in Fig. 2, there are $A+D$ distributions of topics, each of which corresponds to an observed user or source. There are T distributions of words, each corresponds to an unobserved topic. For each document, the generation process repeats for $N_d + N_t$ times where N_d of the iterations correspond to the terms in the document content and N_t corresponds to the terms in the tags. The above again repeats for D times for all documents.

3.3 Model Training

The UCA model includes two sets of unknown parameters, the source-topic distributions θ, and the topic-word distributions ϕ, corresponding to the assignments of individual words to topics z and source x. One can use the Expectation-Maximization (EM) algorithm for estimating parameters in models with latent variables. However, the approach is susceptible to local maxima. In addition, according the posterior probability in Eq. 3, we know the EM will be very computationally expensive due to the sum in the denominator. Thus, we pursue an alternative parameter estimation method, Gibbs sampling [16], which is gaining popularity in topic analysis recently [5, 22]. Instead of estimating the parameters directly, we evaluate the posterior distributions.

While using Gibbs sampling to train generative models, typically, a Markov chain is formed, where the transition between successive states is simulated by repeatedly drawing a topic for each observed term from its conditional probability. The algorithm keeps track of the number of times that a term is assigned to a topic C_{zw}^{TW} and the number of times that a topic is assigned to the user or source $C_{xz}^{(A+D)T}$. Here C^{TW} denotes a $T \times W$ matrix and $C^{(A+D)T}$ denotes a $(A + D) \times T$ matrix, where x, z, ω are the indices of the sources (document or user), topics, and words. We repeat the Gibbs sampling until the perplexity score [3] measured on distributions converges. Algorithm 1 illustrates the Gibbs sampling algorithm for model training.

Algorithm 1 Training User-Content-Annotation Model

1: Given a sequence of triplets $\langle x, d, \omega \rangle$, where d is the document id; ω is the word id; x = nil if ω is a content word; x = user id if ω is a tag word.
2: Given ϵ as the threshold for determining convergence.
3: Initialize C^{TW}, $C^{(A+D)T}$ with random positive values.
4: **repeat**
5: **for all** $\langle x, d, \omega \rangle$ **do**
6: t = z(ω) // get the current topic assignment
7: $C_{tw}^{TW} \leftarrow C_{tw}^{TW} - 1$ //decrement count
8: **if** $x ==$ nil **then**
9: // ω is a document word
10: $C_{dt}^{(A+D)T} \leftarrow C_{dt}^{(A+D)T} - 1$ // decrement count
11: // compute $P(t)$ below
12: **for all** z = 1, ..., T **do**
13: $P(z) \leftarrow P(d, z|\omega) = P(d|z)P(z|\omega)$
14: **end for**
15: sample to obtain t using $P(t)$
16: $C_{dt}^{(A+D)T} \leftarrow C_{dt}^{(A+D)T} + 1$ // increment count
17: **else**
18: // ω is a tag word
19: $C_{xt}^{(A+D)T} \leftarrow C_{xt}^{(A+D)T} - 1$ // decrement count
20: // compute $P(t)$ below
21: **for all** z = 1, ..., T **do**
22: $P(z) \leftarrow P(x, z|\omega) = P(x|z)P(z|\omega)$
23: **end for**
24: sample to obtain t using $P(t)$
25: $C_{xt}^{(A+D)T} \leftarrow C_{xt}^{(A+D)T} + 1$ // increment count
26: **end if**
27: $C_{tw}^{TW} \leftarrow C_{tw}^{TW} + 1$
28: **end for**
29: measure the perplexity on a held-out sample;
30: measure the perplexity change in δ;
31: **until** $\delta \leq \epsilon$

It can be seen from Algo. 1 that the key issue here is the evaluation of the posterior conditional probabilities, i.e. $P(z|w)$, $P(d|z)$, $P(x|z)$, which leads to the evaluation of $P(d|w)$ or $P(x|w)$. Let us again consider the joint probabilities $P(x, z|w)$, $P(d, z|w)$. Similar to earlier work [5, 22], we know the posterior conditional probabilities can be expressed as the product of several conditional probabilities

[3]The measurement of perplexity will be introduced in Sec. 3.4.

on the edges of the Bayesian network. In particular, for documents, we have:

$$P(d,z|\omega) \propto \frac{C_{\omega z}^{WT}+\beta}{\sum_k C_{kz}^{WT}+V\beta}\frac{C_{dt}^{(A+D)T}+\alpha}{\sum_k C_{dk}^{(A+D)T}+T\alpha}, \quad (4)$$

and for users, we have:

$$P(x,z|\omega) \propto \frac{C_{\omega t}^{WT}+\beta}{\sum_k C_{kz}^{WT}+V\beta}\frac{C_{xt}^{(A+D)T}+\alpha}{\sum_k C_{xk}^{(A+D)T}+T\alpha}. \quad (5)$$

Here the unit conditional probabilities in fact are Bayesian estimation of the posteriors: $P(d|z)$, $P(x|z)$ and $P(z|w)$:

$$P(d|z) = \frac{C_{dz}^{(A+D)T}+\alpha}{\sum_k C_{dk}^{(A+D)T}+T\alpha}, \quad (6)$$

$$P(x|z) = \frac{C_{xt}^{(A+D)T}+\alpha}{\sum_k C_{xk}^{(A+D)T}+T\alpha}, \quad (7)$$

$$P(z|\omega) = \frac{C_{\omega t}^{WT}+\beta}{\sum_k C_{kt}^{WT}+V\beta}. \quad (8)$$

Accordingly, for implementation, we need to keep track of $\sum_k C_{dk}^{(A+D)T}$, $\sum_k C_{xk}^{(A+D)T}$ and $\sum_k C_{kt}^{WT}$ in addition to $C_{dt}^{(A+D)T}$, $C_{xt}^{(A+D)T}$ and C_{tw}^{TW}. It is easy to implement these counting using several hash tables. In practice, we set α and β to be $50/T$ and 0.05 respectively. These parameters seem to only affect the convergence of Gibbs sampling but not much the output results, unless the problem is very ill-conditioned.

3.4 Topic Number Selection

The remaining question is how to select the number of topics. We resort to the perplexity measure, which is a standard measure for estimating the performance of a probabilistic model. The perplexity of a set of term-source test pairs, $(\mathbf{w}_d, \mathbf{x}_d)$, for all $d \in D_{test}$ documents, is defined as the exponential of the negative normalized predictive log-likelihood using the trained model:

$$\text{perplexity}(D_{test}) = \exp[-\frac{\sum_{d=1}^{D} \ln P(\mathbf{w}_d|\mathbf{x}_d)}{\sum_{d=1}^{D}|\{\mathbf{w_d},\mathbf{x_d}\}|}]. \quad (9)$$

Here the probability of a set of term-source pairs on a particular document is obtained by a straightforward calculation:

$$P(\mathbf{w}_d|\mathbf{x}_d) = \prod_{(w_d,x_d)\in\{\mathbf{w_d},\mathbf{x_d}\}} P(w_d|x_d) \quad (10)$$

where the probability of an individual term-source pair $P(w_d|x_d)$ is evaluated using the model hierarchy:

$$P(w_d|x_d) = \sum_{t=1}^{T} P(w_d|t)P(t|x_d). \quad (11)$$

Note that the better generalization performance of a model is indicated by a lower perplexity score over a held-out document set. We run the Gibbs sampling using perplexity score as the termination criterion; the topic number is determined by using the smallest T that leads to the near maximum perplexity. Similar approach is also used in previous work for choosing parameters in generative models [2, 17].

4. EXPERIMENTS ON ANNOTATION MODELING

4.1 Data Preparation

A data sample is collected from del.icio.us using the method similar to [20]. We crawled the del.icio.us Web-site starting with a set of popular URL's in Jan. 2006. Then we followed the URL collection of users who have tagged these URL's, arriving at a new set of URL's. By iteratively repeating the above process, we ended up with a collection of 84,961 URL's tagged from May, 1995 to Apr., 2006. There are 9070 users along with 62,007 distinct tag words. Then we crawled the URL's to collect document content. There are 34,530 URL's in the collection which are still valid and have textual content, including 747,935 content words. The activity of users seems to follow a power-law distribution. Since the data we collected is relatively small, many infrequent users and tags might not be included. How to handle resources distributed on the long tail remains an interesting question to explore.

4.2 Topic Number Selection

We first perform the training of the proposed model using the algorithm introduced above. For different settings of the desired topic number, we test the perplexity of the trained model on a held-out sample dataset. Over iterations, the perplexity scores always decreases dramatically after the first several iterations and then soon converges to a stable level. We show a plot of perplexities on five different settings of T in Fig. 3. Here the training set is a 1% random sample of the data available. We are able to see that the larger setting of topic number leads to a lower perplexity score from the start, indicating a better prediction performance. This is because the increased number of topics (before a certain point) reduces the uncertainty in training. For the same reason, the larger setting of topics also leads to a smaller perplexity value in the first several iterations, followed by a sharper drop in perplexity. From the figure, we can see that empirically the algorithm converges within 20 iterations for a relative small sample. For the full dataset, we repeat the Gibbs sampling for 100 iterations.

The second set of experiments carried out seeks to determine the best number of topics in the setting. Using the perplexity measure defined in Eq. 9 - Eq. 11. We perform the experiments by setting different number of topics in training on various sizes of samples from the available data. Generally, the perplexity score first decreases and then remains stable after T is at certain size. We prefer the smallest T that yields a convergence since the greater T requires larger computation. In Fig. 4, we show the perplexity scores over different T for various sample sizes. It is clear that the perplexity decreases much slower from after $T = 80$. Accordingly, we choose the desired topic number to be 80 in the following experiments.

4.3 Discovered Topic Words

We also examine the top words discovered for each topics to judge the quality. Usually the determination of topic words are very subjective and is lack of quantitive measures. Nevertheless, without quantitative assertions, we observe generally high semantic correlations among the top words that are discovered in the same topic. Typically, most discovered topic words are about Web the related applications

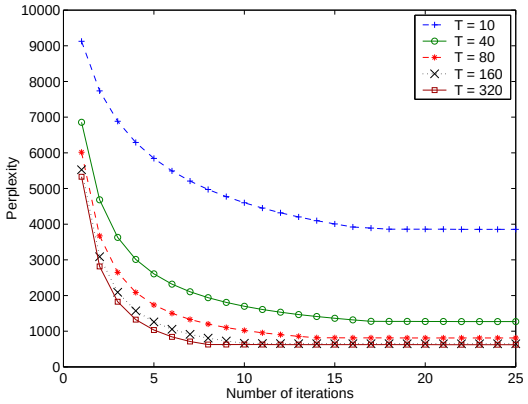

Figure 3: The perplexities over the iterations in training for five different settings of topic number. The training set is a 1% random sample of the available data. The perplexity is tested on a held-out sample whose size is proportional to size of the training set.

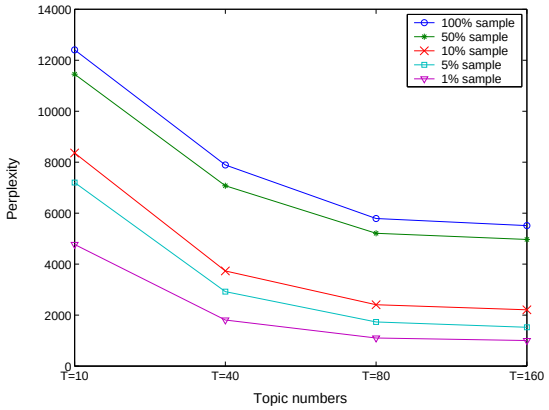

Figure 4: The perplexities over different settings of topic numbers, for $T = 10, 40, 80, 160$. Different sample sizes are tested yielding similar curves indicating a minimum optimal topic number of 80 on the collected data. The perplexity is tested on a held-out sample whose size is proportional to size of the training set.

or softwares. We consider this as a bias of the del.icio.us collection. For Web-based systems with general focus, the topic words can be more sparsely distributed. In addition, we see several cases where some seemingly irrelevant words appear in a relatively coherent word set. But there are few of these cases and the noisy words are usually ranked not high.

Here, we present a subset of the discovered topics due to space limit. As illustrated in Table 1, five topics and their top words are presented. Here, Topic 0 is the topic on Web; Topic 2 is interested in research groups and the programming tools they offer; Topic 3 has many geographical locations; Topic 9 seems to concern about dining and restaurants; Topic 32 is a topic on cooking and kitchen. Note the presented topics do not indicate the distribution of interests – there are many more topics on Web and related online applications.

Topic ID	Top words
0	web site news http information time www page free home software search online text links
2	data work research services group science programming library education file code
3	world states usa country west japan europe north asia australia south russian worldwide
9	product process quality cool sale feedback catalog suggestions patterns pretty rates clothing cds
32	cookies tea sugar cafe orange organic milk bread food egg meat diet fruit kitchen snacks

Table 1: Top words for a selected sample of discovered topics.

5. INFORMATION RETRIEVAL BASED ON RISK MINIMIZATION

In this section, we propose a method to incorporate the topic discovery results discussed in the previous sections into the language modeling-based information retrieval. We first review an information retrieval framework based on Bayesian decision theory. Then, in the next section, we propose a method that naturally combine the topical analysis language models to improve retrieval quality, which is incremental and requires little computational overhead.

In the language modeling (LM) approach to information retrieval (IR), queries and documents are modeled respectively by a probabilistic LM. Let θ_Q denote the parameters of a query model, and let θ_D denote the parameters of a document model. The LM-based IR involves two independent phases: In one case, the generation of a query is viewed as a probabilistic process associated with a certain user. This user first selects the query model θ_Q then picks a query $\mathbf{q}$ from the query model θ_Q with probability $P(\mathbf{q}|\theta_Q)$; In the other case, the document generation has been carried out. First the document language model θ_D is chosen and then the d is generated word by word with probability $P(\mathbf{d}|\theta_D)$. The task of an IR system is to determine the probability of a document being relevant to the query given their LMs are respectively estimated.

Here we work within a risk minimization framework for IR proposed earlier [12]. The framework views the retrieval of relevant documents as those actions to be carried out in Bayesian decision theory. The goal of retrieval is equivalent to minimizing the expected loss.

Risk Minimization Framework: Suppose the relevance is a binary random variable $R \in \{0, 1\}$. Consider the task of a retrieval system as the problem of returning a list of documents to the issued query $\mathbf{q}$. In the general framework of Bayesian decision theory, to each action, there is an associated loss, which, in our case, is the loss for returning a particular document to the user. Assume that the loss function only depends on θ_Q, θ_D, and R_i, the expected risk of returning $\mathbf{d_i}$ is:

$$R(\mathbf{d_i};\mathbf{q}) = \sum_{R\in\{0,1\}} \int_{\Theta_Q}\int_{\Theta_D} L(\theta_Q,\theta_D,R) \times P(\theta_Q|\mathbf{q})P(\theta_D|\mathbf{d_i})P(R|\theta_Q,\theta_D)d\theta_D d\theta_Q \quad (12)$$

where $L(\theta_Q, \theta_D, R)$ is the loss function, $P(\theta_Q|\mathbf{q})$ is the probability of the query model being parameterized by θ_Q given

the query $\mathbf{q}$, $P(\theta_D|\mathbf{d_i})$ is the probability of the document model being parameterized by θ_D given the document $\mathbf{d_i}$, and $P(R|\theta_Q, \theta_D)$ is the probability of relevance of R given the parameter sets are θ_Q and θ_D.

Following earlier work [12], we make the assumption that the loss function only depends on θ_Q and θ_D and is proportional to the distance Δ between θ_Q and θ_D, i.e.,

$$L(\theta_Q, \theta_D, R) \quad \propto \quad \Delta(\theta_Q, \theta_D) \tag{13}$$

The expected risk for returning $\mathbf{d_i}$ to $\mathbf{q}$ is thus:

$$R(\mathbf{d_i}; \mathbf{q}) \propto \int_{\Theta_Q} \int_{\Theta_D} \Delta(\theta_Q, \theta_D) P(\theta_Q|\mathbf{q}) P(\theta_D|\mathbf{d_i}) d\theta_D d\theta_Q. \tag{14}$$

Note here $P(\theta_Q|\mathbf{q})$ depends on the input $\mathbf{q}$ only and is the same for all candidate documents $\mathbf{d_i}$. Rather than explicitly computing the risk in the integral format, we can use the point estimate with the posterior θ_D and θ_D:

$$R(\mathbf{d_i}; \mathbf{q}) \quad \propto \quad \Delta(\widehat{\theta}_q, \widehat{\theta}_{d_i}) P(\theta_D|\mathbf{d_i}). \tag{15}$$

where $\widehat{\theta}_q$ and $\widehat{\theta}_{d_i}$ can be obtained using maximum likelihood estimation observing the words in query and documents.

Further assuming that $P(\theta_D|\mathbf{d_i})$ is the same for all $\mathbf{d_i}$, the risk minimization framework finally becomes a measurement of the distance between two LMs: $\widehat{\theta}_q$ and $\widehat{\theta}_{d_i}$. As in other related work, we can employ the Kullback-Leibler divergence to measure Δ, yielding

$$R(\mathbf{d_i}; \mathbf{q}) \quad \propto \quad \Delta(\widehat{\theta}_q, \widehat{\theta}_{d_i}) \; = \; \sum_w P(w|\widehat{\theta}_q) \log \frac{P(w|\widehat{\theta}_q)}{P(w|\widehat{\theta}_{d_i})}. \tag{16}$$

Comments: According to Eq. 16, the setup of the risk minimization framework has made the measurement of relevance depend only on the LMs of the query and the document, i.e. the posterior parameters $\widehat{\theta}_q$ and $\widehat{\theta}_{d_i}$. This paper proposes a refinement of the query and document LMs using the LMs obtained from social annotations.

6. LANGUAGE MODEL EXPANSION USING SOCIAL ANNOTATIONS

Define our goal now to be improving the LMs of query and documents, say $\widehat{\theta}_q \rightarrow \theta'_q$ and $\widehat{\theta}_{d_i} \rightarrow \theta'_{d_i}$. Here the $\widehat{\theta}_q \rightarrow \theta'_q$ is also known as *query expansion* [11] and the $\widehat{\theta}_{d_i} \rightarrow \theta'_{d_i}$ is also known as *document expansion* [19].

There are several ways for LM expansion. In this paper we focus on the linear interpolation [9] (a.k.a linear smoothing) for combining two LMs. Define an operator $\oplus_\lambda$ for linear smoothing where $a \oplus_\lambda b \equiv \lambda a + (1-\lambda)b$, assuming a, b are both normalized to the same scale. When applied to combining two LMs, θ_1 and θ_2, we define that $\theta_1 \oplus_\lambda \theta_2 \quad \equiv$:

$$\forall v \in \theta_1 \cup \theta_2, \qquad P(v|\theta_1 \oplus_\lambda \theta_2) = \lambda P(v|\theta_1) + (1-\lambda) P(v|\theta_2) \tag{17}$$

where the v here can be a word, a phrase, or simply a token that denotes special meaning (e.g. a topic). In the case when $v \notin \theta_1$, $P(v|\theta_1 \oplus_\lambda \theta_2) = (1-\lambda)P(v|\theta_2)$. Similarly, $P(v|\theta_1 \oplus_\lambda \theta_2) = \lambda P(v|\theta_1)$ when $v \notin \theta_2$. That is, one LM can be easily improved by smoothing with another "better" LM as long as they can be combined using the above linear operator.

Now let us suppose the LMs we want to improve are already estimated. In the following, we give three types of additional LMs we can estimate based on the previous topical analysis of annotations and content. The first model simply treats the annotations as additional terms of the documents; The second model expands the query with the topics; The third model proposes several expansion methods on the document LM.

6.1 Word-Level Annotation Language Model

The annotation LM we give is an ad-hoc improvement. For each document d, let $\tau(d)$ be the set of words in its tags, each having the frequency of being used for d. We are able to estimate a LM, say L_w^d [4], from the observations of $\tau(d)$ for all d's. It easily follows that L_w^d can be combined with $\widehat{\theta}_{d_i}$ using Eq. 17. For Word-level annotation language model, we focus on the simple case of unigram LM, in which each word is assumed to occur depending on the latent probability distribution regardless of the surrounding words.

6.2 Topic-Level Query Language Models

In this and the following section, we seek to make use of the topical analysis on documents previously made in Sec. 3.

Recall in the standard framework, $\widehat{\theta}_q$ is just the empirical distribution of the query $q = \langle w_1, ...w_k \rangle$. This original word-level query model has been shown to underperform [12, 11]. In our approach, we seek to estimate the LMs at higher level. In particular, we consider each topic discovered as a token in the LM. These tokens will later match the topics discovered for the documents to determine their relevance.

First, we estimate the conditional probability that a query word ω belongs to the topic t, say $P(t|w)$. Over all topics, we have a vector $\mathbf{v_{t|w}} = \langle P(t_1|w), ..., P(t_T|w) \rangle$. After normalization, $\mathbf{v_{t|w}}$ becomes the probability distribution over topics, or rather, a topic-level LM.

Second, we merge the multiple topic distributions for each query word into a single topic distribution. Let the desired topic-level query LM be L_t^q. In the unigram case. L_t^q is also a vector of T dimension where each element denotes the probability of a particular topic. Formally, we have:

$$L_t^q = \sum_{w \in q} \delta_w \mathbf{v_{t|w}}. \tag{18}$$

where δ_w is the normalized weight for the word ω, and $L_t^q(i)$ denotes the probability of topic i under this model. Note the setting of δ_w allows us to have $\sum_{i \in L_t^q} L_t^q(i) = 1$. Again, using $\oplus_\lambda$, we combine the models at different levels.

6.3 Topic-Level Document Language Models

Now let us focus on the document LMs. It is easy to see that each document already has a probability distribution over topics discovered from the proposed modeling, denoted by a vector $\mathbf{v_{t|d}} = \langle P(t_1|d), ..., P(t_T|d) \rangle$. Consider this vector as a LM where each topic is a unit. We use $\oplus_\lambda$ to combine this topic-level LM with the original document LM.

[4]Note we use L instead of θ to denote the additional LMs in expansion for clarity. The L_w^d means LM trained at word-level for document expansion. Similarly, the L_t^q indicates the LM at topic level for query expansion.

Then how to leverage the user information in annotations?. Again, recall that the probabilistic model in Sec. 3 also outputs the topic distribution for users. Denote the distribution by a T dimensional vector $\mathbf{u_{t|x}} = \langle P(t_1|x), ..., P(t_T|x)\rangle$. Here each element $P(t_i|x)$ denotes the probability of a user x belonging to the topic t_i. Let the document d be tagged by a set of users, say $U(d)$. We combine the multiple LMs of users in $U(d)$. In particular, the desired model L_t^d is generated in addition to and will be combined with the original topic-level LM of document: $\mathbf{v_{t|d}}$.

Let the trust or importance of user x be δ_x. The L_t^d is obtained as:

$$L_t^d = \delta_d \mathbf{v_{t|d}} + \sum_{x \in U(d)} \delta_x \mathbf{u_{t|x}}, \quad (19)$$

where $\delta_d + \sum_{x \in U(d)} \delta_x = 1$. The δ_d accounts for the emphasis we place on the original discovery of topics for d, and $\forall x \in U(d)$, δ_x determines the trust we place on each user x. Now we have successfully incorporated the topical analysis of documents and users into the original LM-based IR. User domain differences are also considered. How to evaluate user importance is out of the scope of this paper.

7. EXPERIMENTS ON IR QUALITY

7.1 User Domains & Expertise Evaluation

Next we show the probability distribution over topics for several active users. We consider the topics as domains where the users belong to. A higher probability in certain topics indicates stronger interests of this user. For the users with insufficient observations, the domain discovery tends be to less reliable.

Figure 5 illustrates the distributions over 80 topics for three random active users. Users are seperated by their distributions. In general, the overall interest of each user is a mixture of interests in several topics. Some topics for a user is more interesting than others. And for the interested topics, some are more preferred than others.

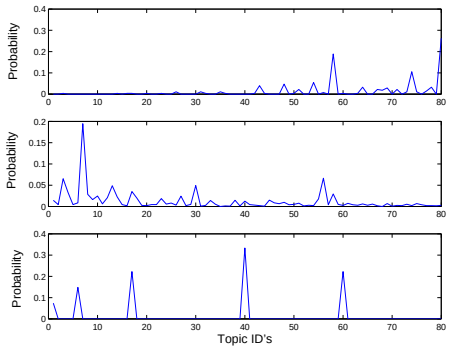

Figure 5: The probability distributions over topics for three active users.

In the following, we discuss the evaluation of user-specific trusts. We start with showing the properties of user activities. Fig. 6 presents the number of authors w.r.t. to the number of tags she has made in the data. It is clear that over 60% of the users contribute less than 50 tags to the data and very few of them make more than 300 words. From the

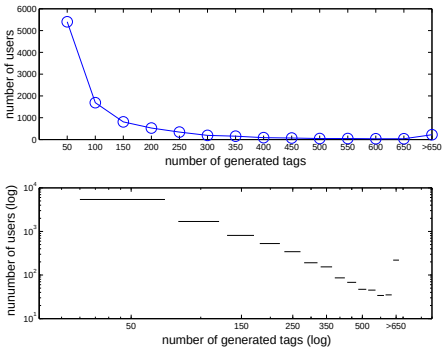

Figure 6: The number of users v.s. number of tags generated, in the normal scale and log-log scale.

log-scale and log-log scale plots, we can see the intensities of user activities follow a power-law distribution.

The power-law property of user activities is in fact helpful for determining the trust we should put on each authors. For most of the cases, users are more or less equivalent in their activity intensity, whom we should not differentiate in the trust scores; For some very active users, we might want to give higher priorities. For simplicity, this paper uses the number of annotations a user has made for the user-specific trust scores. One might consider combine other metrics such as the time duration from last visits or the visit frequencies. Note the framework we propose allows flexible definition of trust scores for users.

7.2 IR Quality

Now let us evaluate the IR quality of various language modeling (LM) approaches. The methods we compare are:

- **Word-level LM on content (W-QD)**: Query LM is trained on the original query and the document LM is trained on the original document content.
- **Word-level LM on content and annotations (W-QDA)**: The query LM is trained on the original query and the document LM is trained on both document content and annotations.
- **Word-level LM + LDA on content and annotations (WT-LDA)**: We run LDA on document plus annotations by treating annotations as additional words, without consideration of user differences. The topic-level LM is combined with W-QDA using the parameter λ_1.
- **Word-level LM + Topic-level LM (WT-QDA)**: We run the proposed topic analysis model on the documents and annotations, obtaining topic information of documents and users. Then, the topic-level LM is combined with the word-level LM W-QDA, using the parameter λ_1.
- **Word-level LM + Topic-level LM on document and users (WT-QDAU)**: User domain interests are considered here. First, the word-level LM and topic-level LM and their combination are trained using WT-QDA. Second, the document LM is combined with the mixture of topics on users who tag the document, using

the parameter λ_2. Note here the users are treated the same in the first step.

- **Word-level LM + Topic-level LM on document, and users with differentiation (WT-QDAU$^+$)**: During the training of the WT-QDAU is obtained using the parameter λ_2, the weights on users are set different.

In addition, we implement the EM-based retrieval method proposed in a related work [20], which is defined as:

- **EM-based information retrieval (EM-IR)**: As proposed in [20], the URL's and users are first clustered using the EM algorithm. Then the probability of seeing certain words for a URL is estimated. Those probabilities are used for retrieval.

For evaluation, we generate 40 queries with lengths varying from one to five words. The words are chosen from tag and document content. Then for each query, we use the above six approaches for document retrieval. The quality of retrieval is evaluated on the top 10 documents using the Discounted Cumulated Gain (DCG) metric [8]. In particular, two human judges are invited to provide feedback on the composite set of URL's which occur in any of the top 10 retrieval results, yielding the DCG_{10} scores. Judgments are carried out independently based on their experience of the relevance quality. Numerical judgment scores of $0, 1, 2$, and 3 are collected to reflect the judges' opinion on the relevance of documents, which respectively imply the sentiment of *poor*, *fair*, *good*, and *perfect*. In general, the judges represent high agreement on the ranking quality. The average judge scores are used for computing the DCG.

In Table 2, we illustrate the DCG_{10} scores for the six approaches: W-QD, EM, W-QDA, WT-LDA, WT-QDA, WT-QDAU, and WT-QDAU$^+$. We can see that both the EM-based IR and the newly proposed approaches outperform the traditional LM-based IR. We read Table 2 from several aspects:

First, we take a look at the improvement according to the use of tags. The EM-based IR proposed in related work [20] increased the DCG scores by 11.5% over traditional LM-based IR (W-QD); The method that uses annotations as additional words improved the DCG by 18.3% (W-QDA over W-QD), which demonstrates that the use of annotation can dramatically improve IR quality.

Second, we examine the improvement based on topical analysis on both document content and annotations. The basic use of the topic information (WT-LDA) further improves the use of annotations (W-QDA) by 2.7%. The topic analysis based on the new generative model, compared with WT-LDA, achieves a gain of 1.3%. It is worthwhile to mention that the LDA-based topic analysis improves a very recent related work [20] (EM-IR) by 9.1%.

Third, we test the improvement by incorporating tagger interests. As illustrated in Table 2, WT-QDAU outperforms pure topic-based IR by 1.1%, showing the importance of user interests.

Fourth, we show the improvement by considering the differences of users while incorporating user interests. The WT-QDAU$^+$ adds another 1.3% in DCG over WT-QDAU. This shows that due to the different user expertise, the quality of tags can be different and thus should be taken into consideration.

W-QD	**EM-IR**	**W-QDA**	**WT-LDA**
7.6192	8.4945	9.0167	9.2602
WT-QDA	**WT-QDAU**	**WT-QDAU$^+$**	
9.3820	9.4938	9.6167	

Table 2: The DCG_{10} scores of six compared approaches: W-QD, EM-IR, W-QDA, WT-LDA, WT-QDA, WT-QDAU, WT-QDAU$^+$.

Overall, the top performance of our proposed model (WT-QDAU$^+$) improved the traditional LM-based IR model by 26%, compared with the the 11.5% improvements by the EM-based approach in [20].

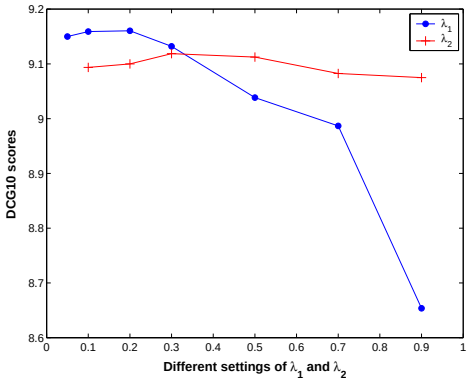

Figure 7: The change of DCG_{10} scores for different settings of λ_1 and λ_2, where λ_1 is the parameter for combining topics with the original LM and λ_2 is the parameter for combining user topic models.

7.3 Sensitivity to Parameter Selection

Finally, we study the effects of parameters in the proposed approach. Two parameters are examined, one being for the WT-QDA (λ_1) and the other for the WT-QDAU (λ_2). Note λ_1 is the weight on the topic-level LM on query and documents and λ_2 is the weight on the LM generated on users.

To determine the optimal λ_1 and λ_2, we perform cross-validation against user judgement. Figure 7 demonstrates the change of DCG scores for different settings of λ_1 and λ_2. From the figure, we can see the proposed approach is very sensitive to λ_1 but less sensitive to λ_2. The λ_1 reaches best performance at around $\lambda_1 = 0.2$. The λ_2 reaches best performances at about $\lambda_2 = 0.3$. This indicates a limited input of topic information will improve LM-based IR but relying on topic information too much fails to differentiate the information to be retrieved.

8. CONCLUSIONS & FUTURE WORK

This paper presents a framework that combines the modeling of information retrieval on the documents associated with social annotations. A new probabilistic generative model is proposed for the generation of document content as well as the associated social annotations. A new way for discovering user domains is presented based on social annotations. Several methods are proposed for combining language models from tags with those from the documents. We then evaluate user expertise based on activity intensities. Experimental evaluation on real-world datasets demonstrates

effectiveness of the proposed model and the improvements over traditional IR approach based on language modeling.

For future work, one could take a closer look at the effects of the parameter sets. It would be useful to reduce the number of parameters for easier tuning for practical use, and focus on exploring more indicators regarding the domain expertise of users and their use in improving user experiences. The inter-personal social networks and communities of users can be more thoroughly studied. How the user social network correlates with social annotations is not clear and remains an interesting question. The temporal dimension of user activities could also be considered on specific queries. In addition, It would be interesting to model the changes in user annotation behaviors. Patterns of the development of user annotations might further advance the use of annotations for more effective information retrieval.

9. ACKNOWLEDGMENTS

We would like to thank David J. Miller from the Department of Electrical Engineering at the Pennsylvania State University for his valuable discussions. This work was funded in part by National Science Foundation and the Raytheon Corporation.

10. REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[3] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, pages 178–186, 2003.

[4] S. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, pages 198–208, 2006.

[5] T. Griffiths and M. Steyvers. Finding scientific topics. In *National Academy of Sciences*, 2004.

[6] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, June 2006. Springer.

[7] P. Jackson. *Introduction to expert systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.

[8] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval*, pages 41–48, 2000.

[9] F. Jelinek and R. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Pattern recognition in Practice*, 1980.

[10] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *Proceedings of the 12th international conference on World Wide Web*, pages 568–576, 2003.

[11] O. Kurland, L. Lee, and C. Domshlak. Better than the real thing?: iterative pseudo-query processing using cluster-based language models. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2005. ACM Press.

[12] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international conference on Research and development in information retrieval*, pages 111–119, 2001.

[13] A. K. McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI'09 Workshop on Text Learning*, 1999.

[14] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 198–207, New York, NY, USA, 2005. ACM Press.

[15] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM Press.

[16] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Publisher, 2nd Edition, 2005.

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

[18] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315. ACM Press, 2004.

[19] T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *HLT-NAACL*, 2006.

[20] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2006. ACM Press.

[21] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transaction of Information System*, 22(2):179–214, 2004.

[22] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 173–182. ACM Press, 2006.