Automatic Labeling of Multinomial Topic Models

Qiaozhu Mei, Xuehua Shen, Chengxiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana,IL 61801
{qmei2,xshen,czhai}@uiuc.edu

ABSTRACT

Multinomial distributions over words are frequently used to model topics in text collections. A common, major challenge in applying all such topic models to any text mining problem is to label a multinomial topic model accurately so that a user can interpret the discovered topic. So far, such labels have been generated manually in a subjective way. In this paper, we propose probabilistic approaches to automatically labeling multinomial topic models in an *objective* way. We cast this labeling problem as an optimization problem involving minimizing Kullback-Leibler divergence between word distributions and maximizing mutual information between a label and a topic model. Experiments with user study have been done on two text data sets with different genres. The results show that the proposed labeling methods are quite effective to generate labels that are meaningful and useful for interpreting the discovered topic models. Our methods are general and can be applied to labeling topics learned through all kinds of topic models such as PLSA, LDA, and their variations.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Text Mining

General Terms: Algorithms

Keywords: Statistical topic models, multinomial distribution, topic model labeling

1. INTRODUCTION

Statistical topic modeling has attracted much attention recently in machine learning and text mining [11, 4, 28, 22, 9, 2, 16, 18, 14, 24] due to its broad applications, including extracting scientific research topics [9, 2], temporal text mining [17, 24], spatiotemporal text mining [16, 18], author-topic analysis [22, 18], opinion extraction [28, 16], and information retrieval [11, 27, 25]. Common to most of this work is the idea of using a multinomial word distribution (also called a unigram language model) to model a topic in text.

For example, the multinomial distribution shown on the left side of Table 1 is a topic model extracted from a collection of abstracts of database literature. This model gives high probabilities to words such as "view", "materialized", and "warehouse," so it intuitively captures the topic "materialized view." In general, a different distribution can be regarded as representing a different topic.

Many different topic models have been proposed, which can extract interesting topics in the form of multinomial distributions automatically from text. Although the discovered topic word distributions are often intuitively meaningful, a major challenge shared by all such topic models is to accurately interpret the meaning of each topic. Indeed, it is generally very difficult for a user to understand a topic merely based on the multinomial distribution, especially when the user is not familiar with the source collection. It would be hard to answer questions such as "What is a topic model about?" and "How is one distribution different from another distribution of words?".

Without an automatic way to interpret the semantics of topics, in existing work of statistical topic modeling, people generally either select top words in the distribution as primitive labels [11, 4, 9, 2], or generate more meaningful labels manually in a subjective manner [17, 16, 18, 24]. However, neither of these options is satisfactory. Consider the following topic extracted from a collection of database literature:

Topic Model		**Variant Labels**
views	0.10	**Top Terms**: views, view, materialized,
view	0.10	maintenance, warehouse, tables
materialized	0.05	**Human**: materialized view, data warehouse
maintenance	0.05	
warehouse	0.03	**Single Term**: view, maintenance;
tables	0.02	**Phrase**: data warehouse, view maintenance
summary	0.02	**Sentence**: Materialized view selection and
updates	0.02	maintenance using multi-query optimization

Table 1: Variant possible labels for a topic model

It is difficult for someone not familiar with the database domain to infer the meaning of the topic model on the left just from the top terms. Similar examples can be found in scientific topics, where extracting top terms is not very useful to interpret the coherent meaning of a topic. For example, a topic labeled with "insulin glucose mice diabetes hormone"[1] may be a good topic in medical science, but makes little sense to common audience.

Manual labeling also has its own problems. Although manually generated labels are usually more understandable and better capture the semantics of a topic (see Table 1), it requires a lot of human effort to generate such labels.

[1] www.cs.cmu.edu/∼lemur/science/topics.html, Topic 26

KDD'07, August 12–15, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-609-7/07/0008 ...$5.00.

A more serious problem with manual labeling is that the labels generated are usually *subjective* and can easily be biased towards the user's personal opinions. Moreover, relying on human labeling also makes it hard to apply such topic models to online tasks such as summarizing search results.

Thus it is highly desirable to automatically generate meaningful labels for a topic word distribution so as to facilitate interpretations of topics. However, to the best of our knowledge, no existing method has been proposed to automatically generate labels for a topic model or a multinomial distribution of words, other than using a few top words in the distribution to label a topic. In this paper, we study this fundamental problem which most statistical topic models suffer from and propose probabilistic methods to automatically label a topic.

What makes a good label for a topic? Presumably, a good label should be understandable to the user, could capture the meaning of the topic, and distinguish a topic from other topics. In general, there are many possible choices of linguistic components as topic labels, such as single terms, phrases, or sentences. However, as we could learn from Table 1, single terms are usually too general and it may not be easy for a user to interpret the combined meaning of the terms. A sentence, on the other hand, may be too specific, thus it could not accurately capture the *general* meaning of a topic. In between these two extremes, a phrase is coherent and concise enough for a user to understand, while at the same time, it is also broad enough to capture the overall meaning of a topic. Indeed, when labeling topic models manually, most people prefer phrases [17, 16, 18, 24]. In this paper, we propose a probabilistic approach to automatically labeling topic models with meaningful phrases.

Intuitively, in order to choose a label that captures the meaning of a topic, we must be able to measure the "semantic distance" between a phrase and a topic model, which is challenging. We solve this problem by representing the semantics of a candidate label with a word distribution and casting this labeling problem as an optimization problem involving minimizing the Kullback-Leibler divergence between the topic word distribution and a candidate label word distribution, which can be further shown to be maximizing mutual information between a label and a topic model.

The proposed methods are evaluated using two text data sets with different genres (i.e., literature and news). The results of experiments with user study show that the proposed labeling methods are quite effective and can automatically generate labels that are meaningful and useful for interpreting the topic models.

Our methods are general and can be applied to labeling a topic learned through all kinds of topic models such as PLSA, LDA, and their variations. Indeed, it can be applied as a post-processing step to any topic model, as long as a topic is represented with a multinomial distribution over words. Moreover, the use of our method is not limited to labeling topic models; our method can also be used in any text management tasks where a multinomial distribution over words can be estimated, such as labeling document clusters and summarizing text. By switching the context where candidate labels are extracted and where the semantic distance between a label and a topic is measured, we can use our method to generate labels that can capture the content variation of the topics over different contexts, allowing us to interpret topic models from different views. Thus our labeling methods also provide an alterative way of solving a major task of contextual text mining [18].

The rest of the paper is organized as follows. In Section 2, we formally define the problem of labeling multinomial topic models. In Section 3, we propose our probabilistic approaches to generating meaningful phrases as topic labels. The variation of this general method is discussed in Section 4, followed by empirical evaluation in Section 5, discussion of related work in Section 6, and our conclusions in Section 7.

2. PROBLEM FORMULATION

Given a set of latent topics extracted from a text collection in the form of multinomial distributions, our goal is, informally, to generate understandable semantic labels for each topic. We now formally define the problem of topic model labeling. We begin with a series of useful definitions.

Definition 1 (Topic Model) A *topic model* θ in a text collection $\mathcal{C}$ is a probability distribution of words $\{p(w|\theta)\}_{w\in V}$ where V is a vocabulary set. Clearly, we have $\sum_{w\in V} p(w|\theta) = 1$.

Intuitively, a topic model can represent a semantically coherent topic in the sense that the high probability words often *collectively* suggest some semantic theme. For example, a topic about "SVM" may assign high probabilities to words such as "supporting", "vector" and "kernel." It is generally assumed that there are multiple such topic models in a collection.

Definition 2 (Topic Label) A *topic label*, or a "*label*", l, for a topic model θ, is a sequence of words which is semantically meaningful and covers the latent meaning of θ.

Words, phrases, and sentences are all valid labels under this definition. In this paper, however, we only use phrases as topic labels.

For the example above, a reasonable label may be "supporting vector machine."

Definition 3 (Relevance Score) The *relevance score* of a label to a topic model, $s(l, \theta)$, measures the semantic similarity between the label and the topic model. Given that l_1 and l_2 are both meaningful candidate labels, l_1 is a better label for θ than l_2 if $s(l_1, \theta) > s(l_2, \theta)$.

With these definitions, the problem of **Topic Model Labeling** can be defined as follows:

Given a topic model θ extracted from a text collection, the problem of *single topic model labeling* is to (1) identify a set of candidate labels $L = \{l_1, ..., l_m\}$, and (2) design a relevance scoring function $s(l_i, \theta)$. With L and s, we can then select a subset of n labels with the highest relevance scores $L_\theta = \{l_{\theta,1}, ..., l_{\theta,n}\}$ for θ.

This definition can be generalized to label multiple topics. Let $\Theta = \{\theta_1, ..., \theta_k\}$ be a set of k topic models, and $L = \{l_1, ..., l_m\}$ be a set of candidate topic labels. The problem of *multiple topic model labeling* is to select a subset of n_i labels, $L_i = \{l_{i,1}, ..., l_{i,n_i}\}$, for each topic model θ_i. In most text mining tasks, we would need to label multiple topics.

In some scenarios, we have a set of well accepted candidate labels (e.g., the Gene Ontology entries for biological topics). However, in most cases, we do not have such a candidate set. More generally, we assume that the set of candidate labels can be extracted from a reference text collection, which is related to the meaning of the topic models. For example, if the topics to be labeled are research themes in data mining, the reasonable labels could be extracted from the

KDD conference proceedings. In most text mining tasks, it would be natural to use the text collection to be mined as our reference text collection to extract candidate labels.

Therefore, a natural work flow for solving the topic labeling problem would be (1) extracting a set of candidate labels from a reference collection; (2) finding a good relevance scoring function; (3) using the score to rank candidate labels w.r.t. each topic model; and (4) select top ranked ones to label the corresponding topic.

However, many challenges need to be solved in order to generate good topic labels automatically. As discussed in Section 1, a good set of labels for a topic should be (1) understandable, (2) semantically relevant, (3) covering the whole topic well, and (4) discriminative across topics. Without prior domain knowledge, extracting understandable candidate labels is non-trivial. Since a topic model and a label have different representations, it is also difficult to compare their semantics. As a result, there is no existing method to measure the semantic relevance between a topic model and a label. Even with a good measure for semantic relevance, it is still unclear how we can ensure that the label would fully cover the meaning of a topic and also capture the difference between different topic models.

In the next section, we propose a probabilistic approach to generating labels for topic models automatically.

3. PROBABILISTIC TOPIC LABELING

To generate labels that are ***understandable***, semantically ***relevant***, ***discriminative*** across topics, and of ***high coverage*** of each topic, we first extract a set of understandable candidate labels in a preprocessing step, then design a relevance scoring function to measure the semantic similarity between a label and a topic, and finally propose label selection methods to address the inter-topic discrimination and intra-topic coverage problems.

3.1 Candidate Label Generation

As discussed in Section 1, compared with single terms and sentences, phrases appear to be more appropriate for labeling a topic. Therefore, given a reference collection $\mathcal{C}$, the first task is to generate meaningful phrases as candidate labels. Phrase generation has been addressed in existing work [7, 26, 15, 6]. In general, there are two basic approaches:

Chunking/Shallow Parsing: Chunking (Shallow Parsing) is a common technique in natural language processing, which aims at identifying short phrases, or "chunks" in text. A chunker often operates on text with part of speech tags, and uses the tags to make decisions of chunking according to some grammar, or through learning from labeled training sets. In our work, we extract the chunks/phrases frequently appearing in the collection as candidate labels.

The advantage of using an NLP chunker is that the phrases generated are grammatical and meaningful. However, the accuracy of chunking usually depends heavily on the domain of the text collection. For example, if the model is trained with news articles, it may not be effective on scientific literature. Even in scientific literature, processing biology literature is much different from processing computer science publications.

Ngram Testing: Another type of method is to extract meaningful phrases from word ngrams based on statistical tests. The basic idea is that if the words in an ngram tend to co-occur with each other, the ngram is more likely to be an n-word phrase.

There are many methods for testing whether an ngram is a meaningful collocation/phrase [7, 26, 1, 15]. Some methods rely on statistical measures such as mutual information [7]. Others rely on hypothesis testing techniques. The null hypothesis usually assumes that "the words in an ngram are independent", and different test statistics have been proposed to test the significance of violating the null hypothesis. Two famous hypothesis testing methods showing good performance on phrase extraction are χ^2 Test and Student's T-Test [15].

The advantage of such an ngram testing approach is that it does not require training data, and is applicable to text collection of any ad hoc domains/topics. The disadvantage is that the top ranked ngrams sometimes are not linguistically meaningful, and it usually only works well for bigrams.

In our experiments, we compare both approaches to extract the set of candidate labels.

3.2 Semantic Relevance Scoring

We propose two relevance scoring functions to rank labels by their semantical similarity to a topic model.

3.2.1 *The Zero-Order Relevance*

The semantics of a latent topic θ is fully captured by the corresponding multinomial distribution. Intuitively any reasonable measure of the semantic relevance of a label to a topic should compare the label with this distribution in some way.

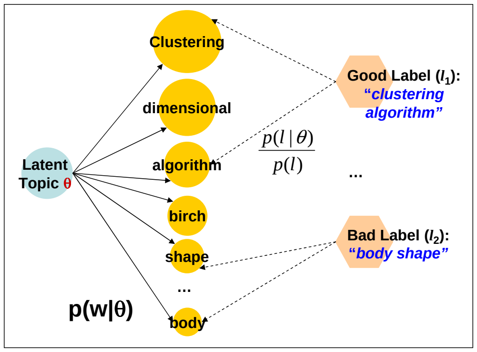
Clustering
dimensional
algorithm
birch
shape
...
body
Latent Topic θ
p(w|θ)
Good Label (l_1): "clustering algorithm"
$\frac{p(l|\theta)}{p(l)}$
...
Bad Label (l_2): "body shape"

Figure 1: Illustration of zero-order relevance
A larger circle means a higher probability.

One possibility is to define the semantic relevance score of a candidate phrase $l = u_0 u_1 ... u_m$ (u_i is a word) as

$$Score = \log \frac{p(l|\theta)}{p(l)} = \sum_{0 \le i \le m} \log \frac{p(u_i|\theta)}{p(u_i)}$$

where the independence of $u_i' s$ is assumed. The basic idea of this approach is illustrated in Figure 1. Basically, a phrase containing more "important" (high $p(w|\theta)$) words in the topic distribution is assumed to be a good label. $p(u_i)$ is to correct the bias toward favoring short phrases and can be estimated using some background collection B, or simply set to uniform. With this method, we essentially score a candidate phrase based on the likelihood that the phrase is "generated" using the topic model θ as opposed to some background word distribution.

We say that this method captures the "zero-order relevance" since no context information from the reference collection is considered. Although this method is simple and intuitively easy to understand, the semantic information of the label is ignored and the information carried by the entire topic distribution is not fully utilized. A highly ranked label may happen to consist of many high probability words but have quite different meaning from the topic. A topic in computer science containing "tree" and "apple" may not be about "apple tree". We now propose another method based on deeper analysis of semantics.

3.2.2 The First-order Relevance

The semantics of a topic model should be interpreted in a context. For example, a topic about "rule", "association", "correlated", "frequency" is difficult to be labeled without a context of data mining. To "decode" the meaning of the topic conveyed by a multinomial distribution, a suitable context should be considered. In such a context, terms with higher probabilities in the distribution are more likely to appear when the topic θ is covered in a document. A natural context to interpret a topic is the original collection from which the topic model is extracted.

As discussed in Section 2, one challenge in topic labeling is the mismatch of the representation of a topic model and a label. Our idea is thus to represent a candidate label also with a multinomial distribution of words, which we can then use to compare with the topic model distribution to decide whether the label and the topic have the same meaning. Ideally, let us assume that there is also a multinomial distribution $\{p(w|l)\}$ decided by label l. We can measure the closeness of $\{p(w|l)\}$ and $\{p(w|\theta)\}$ using the Kullback-Leibler(KL) divergence $D(\theta||l)$. Intuitively, this KL divergence can capture how good l is as a label for θ. If l is a perfect label for θ, these two distributions should perfectly match each other, thus the divergence would be zero. The basic idea of this method is illustrated in Figure 2.

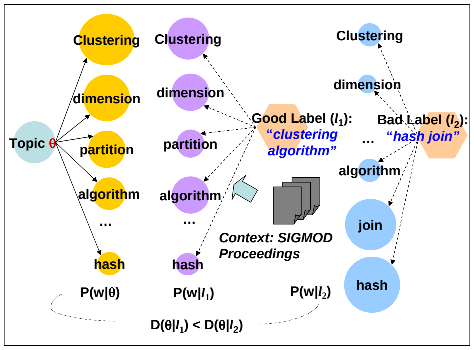

Figure 2: Illustration of first order relevance
A larger circle means a higher probability.

Unfortunately, there is no clue about this unknown distribution $\{p(w|l)\}$. To use this relevance score, we thus would need to approximate $\{p(w|l)\}$. One way to approximate $\{p(w|l)\}$ is to include a context collection $\mathcal{C}$, and estimate a distribution $\{p(w|l,\mathcal{C})\}$ to substitute $\{p(w|l)\}$. This approximation is reasonable: Consider the scenario when a person is unfamiliar with the meaning of a phrase, he/she would look at the context of the phrase first, and then decide whether the phrase is good to label a topic. For example, as in Figure 2, to label a database research topic, a reasonable context could be the SIGMOD conference proceedings. "Clustering algorithm" is a much better label for θ than "hash join" is, because the multinomial distribution estimated based on the context of "clustering algorithm" better matches the topic distribution θ than that based on the context of "hash join." We refer to the reference collection $\mathcal{C}$ as the ***context*** of topic model labeling.

3.2.3 Relevance Scoring Function

Formally, the relevance scoring function of label l w.r.t. topic model θ is defined as the negative KL divergence of $\{p(w|\theta)\}$ and $\{p(w|l)\}$. With the introduction of the context $\mathcal{C}$, this scoring function can be rewritten as follows:

$$\begin{aligned}
Score(l,\theta) &= -D(\theta||l) = -\sum_w p(w|\theta)\log\frac{p(w|\theta)}{p(w|l)} \\
&= -\sum_w p(w|\theta)\log\frac{p(w|\mathcal{C})}{p(w|l,\mathcal{C})} - \sum_w p(w|\theta)\log\frac{p(w|\theta)}{p(w|\mathcal{C})} \\
&\quad -\sum_w p(w|\theta)\log\frac{p(w|l,\mathcal{C})}{p(w|l)} \\
&= \sum_w p(w|\theta)\log\frac{p(w,l|\mathcal{C})}{p(w|\mathcal{C})p(l|\mathcal{C})} - D(\theta||\mathcal{C}) \\
&\quad -\sum_w p(w|\theta)\log\frac{p(w|l,\mathcal{C})}{p(w|l)} \\
&= \sum_w p(w|\theta)PMI(w,l|\mathcal{C}) - D(\theta||\mathcal{C}) + Bias(l,\mathcal{C})
\end{aligned}$$

From this rewriting, we see that the scoring function can be decomposed into three components. The second component is the KL divergence between the topic and the labeling context. Intuitively, if we use humanity literature as the context to label a data mining topic, the relevance score will be lower since it is not as trustworthy as computer science literature. However, this divergence is identical for all candidate labels, thus can be ignored in ranking labels. The third component can be viewed as a bias of using context $\mathcal{C}$ to infer the semantic relevance of l and θ. Consider the scenario that l is a good label for θ according to some prior knowledge, such as a domain ontology, but does not appear in $\mathcal{C}$. In this case, $\mathcal{C}$ is biased to be used to infer the semantics of l w.r.t. θ. Practically, $Bias(l,\mathcal{C})$ can be utilized to incorporate priors of candidate labels. When both the topic models and the candidate labels are generated from the collection $\mathcal{C}$, we simply assume that there is no bias.

Interestingly, the first component can be written as the expectation of pointwise mutual information between l and the terms in the topic model given the context $(E_\theta(PMI(w,l|\mathcal{C})))$. Without any prior knowledge on the label-context bias, we rank the candidate labels with $E_\theta(PMI(w,l|\mathcal{C}))$, where $PMI(w,l|\mathcal{C})$ can all be pre-computed independently of the topic models to be labeled.

Note that $PMI(w,l|\mathcal{C})$ is undefined if $p(w,l|\mathcal{C}) = 0$. One simple strategy is to ignore such w in the summation. A more reasonable way is to smooth $p(w,l|\mathcal{C})$ with methods like Laplace smoothing.

This relevance function is called the first-order relevance of a label to a topic.

3.2.4 Intuitive Interpretation

Ranking candidate labels based on $E_\theta(PMI(w,l|\mathcal{C}))$ is technically well motivated. However, is this a reasonable formalization in reality? What does this ranking function essentially capture? In this section, we give an intuitive interpretation of this semantic relevance scoring function.

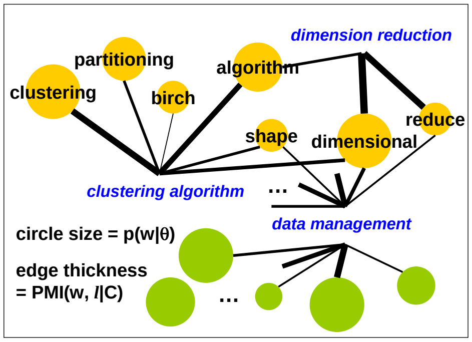

Figure 3: Interpretation of label selection
A label having high PMI with many high probability topic words would be favored.

As shown in Figure 3, we can construct a weighted graph, where each node is either a term in a topic model (weighted with $\{p(w|\theta)\}$) or a candidate label. Each edge between a label and a topical term is then weighted with the pointwise mutual information $PMI(w, l|\mathcal{C})$, which is often used to measure semantic associations [7, 15, 20]. Thus the weight of each node indicates the importance of the term to this topic, while the weight of each edge indicates how strongly the label and the term are semantically associated.

The scoring function $E_\theta(PMI(w, l|\mathcal{C}))$ would rank a label node higher, if it generally has *stronger semantic relation* (thicker edge) to those *important topical words* (larger circles). Intuitively, the labels selected in this way are meant to cover the entire topic model well.

3.3 High Coverage Labels

The third criterion of a good label is the high intra-topic coverage. We expect a label to cover as much semantic information of a topic as possible. Indeed, if we only extract one label for each topic, the semantic relevance function already guarantees that the the label covers maximum semantic information of θ. However, one label usually only partially covers a topic. When selecting multiple labels, we naturally expect the new labels to cover different aspects of the topic, not the information covered by the labels already selected.

Intuitively, in Figure 3, let us assume that "clustering algorithm" is already selected to label the upper topic. However, there are still important topical nodes (e.g., "dimensional", "reduce") weakly covered, or not covered by the label. We thus expect the second label to cover this missing information as much as possible, thus we prefer "dimension reduction", rather than labels like "clustering technique".

To implement this intuition, we propose to select labels with the Maximal Marginal Relevance (MMR) [5] criterion. MMR is commonly used in information retrieval tasks, where both high relevance and low redundancy of retrieval results are desired.

Specifically, we select labels one by one, by maximizing the MMR criterion when selecting each label:

$$\hat{l} = \arg\max_{l \in L-S}[\lambda Score(l, \theta) - (1-\lambda)\max_{l' \in S} Sim(l', l)]$$

where S is the set of labels already selected, $Sim(l', l) = -D(l'||l) = -\sum_w p(w|l')\log\frac{p(w|l')}{p(w|l)}$, and λ is a parameter to be empirically set.

3.4 Discriminative Labels

The previous criteria all only consider the labeling of a single topic. When a set of topics are presented, achieving inter-topic discrimination would be another criterion to consider. A label with high relevance scores to many topic models would not be very useful in this case even though it may be a good label for each individual topic. Intuitively, in Figure 3, "clustering algorithm" is a better label for the upper topic than "data management", since the former covers the important nodes well and exclusively.

In principle, we expect a good label to have high semantic relevance to the target topic model, and low relevance to other topic models. We thus propose the following modified scoring function:

$$Score'(l, \theta_i) = Score(l, \theta_i) - \mu Score(l, \theta_{1,\ldots,i-1,i+1,\ldots,k})$$

where $\theta_{1,\ldots,i-1,i+1,\ldots,k}$ (short as θ_{-i}) is the semantics carried by all other topics than θ_i, and μ controls the discriminative power. $Score(l, \theta_{-i})$ can be modeled as

$$\begin{aligned}
Score(l, \theta_{-i}) &= -D(\theta_{-i}||l) \\
&\overset{\text{rank}}{=} E_{\theta_{-i}}(PMI(w, l|\mathcal{C})) \\
&\approx \frac{1}{k-1}\sum_{j=1,..,i-1,i+1,..,k}\sum_w p(w|\theta_j)(PMI(w, l|\mathcal{C})) \\
&= \frac{1}{k-1}\sum_{j=1..k} E_{\theta_j}(PMI(w, l|\mathcal{C})) \\
&\quad -\frac{1}{k-1}E_{\theta_i}(PMI(w, l|\mathcal{C}))
\end{aligned}$$

which leads to

$$Score'(l, \theta_i) \approx$$

$$(1+\frac{\mu}{k-1})E_{\theta_i}(PMI(w, l|\mathcal{C})) - \frac{\mu}{k-1}\sum_{j=1..k}E_{\theta_j}(PMI(w, l|\mathcal{C}))$$

We use $Score'(l, \theta)$ to rank the labels, which achieves the needed discrimination across topic models.

With the methods proposed in this section, we are able to generate labels for multinomial topic models, which are understandable, semantically relevant, discriminative across topics, and of high coverage inside topics.

Although it is motivated to label multinomial topic models, the use of our approach is not limited to this. Variations in using the labeling approach could lead to different interesting applications. In the following section, we present two possible applications of topic model labeling.

4. VARIATIONS OF TOPIC LABELING

In the previous section, we proposed the probabilistic framework and methods to label topic models, in which we assume that there is a multinomial representation for each topic model, and a context collection to generate candidate labels and measure the semantic relevance of a candidate label to a topic. In this section, we relax the assumption and introduce some variations of topic labeling, which can lead to many interesting text mining applications.

4.1 Labeling Document Clusters

The topic labeling framework, which is proposed to label topic models, essentially consists of a multinomial word distribution, a set of candidate labels, and a context collection. Thus it could be applied to any text mining problems, in which a multinomial distribution of word is involved.

In some tasks, such as topic modeling and many information retrieval tasks [8, 27], a multinomial word distribution is explicit. In other tasks, however, a multinomial word distribution may not be directly available; in such tasks, we can also apply our method by extracting a multinomial distribution. For example, given a group of documents $G = \{d_1, ..., d_m\}$, a multinomial word distribution can be easily constructed using the maximum likelihood estimation:

$$p_G(w) = \frac{\sum_{d \in G} c(w, d)}{\sum_{d' \in G} \sum_{w'} c(w', d')}$$

The proposed multinomial topic model labeling methods can be easily applied to generating labels for $\{p_G(w)\}$. Such labels can thus be used to interpret the original group of documents. This is extremely valuable as many text management tasks involve a group/groups of documents, whose latent semantics is difficult to present. For example, document clustering partitions a collection of documents into groups, where a good label for each group may help the user understand why these documents are grouped together.

Labeling a cluster of documents is also valuable for many other tasks, such as search result summarization and model-based feedback [27]. In fact, the topic labeling method can be applied to any mining problems where a multinomial distribution of words can be estimated, such as term clustering, annotation of frequent patterns in text.

4.2 Context Sensitive Labeling

Another possible variation is to switch the context collection, i.e., label a topic model extracted from one collection with another collection as the context. Although we normally would like to label a topic using the collection from which the topic is extracted as the context, it may be interesting sometimes to label/interpret a topic model in different contexts. Such cross-context interpretation can help us understand the variations of a topic and the connections between different contexts. For example, interpreting a topic discovered from one research area (e.g., database) in the context of another related research area (e.g., information retrieval) may reveal interesting connections between the two areas (e.g., an interdisciplinary research theme). Since our method can work on any context, we can easily use it to achieve such cross-context interpretation of topics, and contextual text mining in general.

Indeed, a major task of contextual text mining is to extract topics and compare their variations in different contexts (e.g., time [17], location [16], authorship [22, 18], etc). In the existing work, this is done by designing a specific statistical topic model with contextual structure, and fitting the data directly with the model. Topic labeling provides an alternative way to track the context-sensitive semantics of a general topic. By using different context collections, the semantics of candidate labels and topic models are biased towards the context. The labeling algorithm thus can generate different labels for the same topic, from the view in different contexts. Such technique can be applied to many contextual text mining tasks, such as temporal, spatiotemporal text mining, and author-topic analysis.

In Section 5, we show that variations of the general topic labeling framework are effective for different mining tasks.

5. EXPERIMENTS AND RESULTS

In this section, we present the results of our evaluation of the effectiveness of the proposed methods for automatically labeling multinomial topic models using two data sets.

5.1 Experiment Setup

Data Sets: We explore two different genres of document collections: the SIGMOD conference proceedings, and the Associated Press (AP) news dataset. To construct the first dataset, we downloaded 1848 abstracts of SIGMOD proceedings between the year 1975 and 2006, from the ACM digital library[2]. The second data collection contains a set of 2246 AP news articles, downloaded from http://www.cs.princeton .edu/∼blei/lda-c/ap.tgz. We built an index for each collection and implemented the topic labeling methods proposed in Section 3 with the Lemur toolkit[3].

Candidate Labels: We generate two sets of candidate labels with different methods: (1) extract noun phrases chunked by an NLP Chunker[4]; (2) extract most significant 2-grams using the N-gram Statistics Package [1]. We use the T-Test to test the significance of 2-grams, and extract those with the highest T-Scores [15]. More specifically, we extract the top 1000 candidate 2-grams ranked by T-Score and top 1000 chunked noun phrases ranked by their frequencies. The ngrams with the highest T-Scores and the most frequent noun phrases are presented in Table 2.

SIGMOD		AP	
2-gram	noun phrase	2-gram	noun phrase
database systems	this paper	he said	the united states
database system	the problem	more than	the government
object oriented	a set	united states	last year
query processing	the data	new york	the country
data base	the database	last year	the nation

Table 2: Sample candidate labels

Topic Models: From each dataset, we extract a number of topics using two representative statistical topic models, the PLSA [11] and LDA [4]. A background component model is added into PLSA to absorb the non-informative words, as suggested in [28] and [17]; this will make the topic models more distinguishable and readable. We do not use such a background model, or prune stopwords for LDA, in order to test the robustness of our topic labeling methods. We extracted 30 and 50 major topics from the SIGMOD and AP dataset, respectively. A subset of example topics is shown in Table 3, where we list the words of the highest probabilities for each topic in the bottom row. We can see that for some topics, especially those from news articles (AP), it is hard to tell the latent meaning merely from the top words.

5.2 Effectiveness of Topic Labeling

We first show some sample results of our topic labeling method in Table 3; for comparison, we also show the human-generated labels for the same topics. It is clear that the automatically generated labels can all capture the meaning of the topic to some extent; indeed, most of them are as good as human generated labels (e.g., "clustering algorithm" and "data streams"), though some are not (e.g., "air force"). Some topics are difficult to interpret even by human (e.g., "death sentence").

To quantitatively evaluate the effectiveness of the automatic labeling methods, we ask three human assessors to compare the results generated by different methods. Specifically, for each of the most salient topics generated with PLSA (12 topics from SIGMOD and 18 topics from AP), we present to the annotators the labels generated by different methods in a random order, together with the word

[2] http://www.acm.org/dl

[3] http://www.lemurproject.org/

[4] http://opennlp.sourceforge.net/

	SIGMOD				AP			
Auto Label	**clustering algorithm**	**r tree**	**data streams**	**concurrency control**	**air force**	**court appeals**	**dollar rates**	**iran contra**
Man. Label	**clustering algorithms**	**indexing methods**	**Stream data management**	**transaction management**	**air plane crash**	**death sentence**	**international stock trading**	**iran contra trial**
θ	clustering	tree	stream	transaction	plane	court	dollar	north
	clusters	trees	streams	concurrency	air	judge	1	case
	video	spatial	continuous	transactions	flight	attorney	yen	trial
	dimensional	b	monitoring	recovery	pilot	prison	from	iran
	cluster	r	multimedia	control	crew	his	late	documents
	partitioning	disk	network	protocols	force	trial	gold	walsh
	quality	array	over	locking	accident	case	down	reagan
	birch	cache	ip	log	crash	convicted	london	charges

Table 3: Sample topics and system-generated labels
The second row contains the automatically generated labels. The third row presents the manually generated labels. The fourth row shows the words of highest probabilities in the topic distribution.

distribution and the most relevant documents to this topic to help a human assessor interpret the topic. A baseline method is included in comparison, which simply uses the top k terms in the word distribution as the topic labels. The other methods included in the comparison are shown in Table 4.

System	Cand. Labels	Relevance Score
NGram-1	Ngrams	First-order
NGram-0-U	Ngrams	0-order, uniform normaliz.
NGram-0-B	Ngrams	0-order, norm. with $p(w\|B)$.
Chunk-1	NP Chunks	First-order

Table 4: Systems Compared in Human Evaluation
Default parameter setting: $\lambda = 0.2$, $\mu = 0.7$ for AP; $\lambda = 0.2$, $\mu = 1$ for SIGMOD

Given the labels generated by n (n = 2, 3...) systems, we ask the assessors to rank the systems according to the quality of the labels they generated. For each topic, they will assign a score of $n - k$ to a system if it is ranked at the k'th place. If the labels from several systems are difficult to be distinguished/ranked, we first give them an arbitrary ranking, and then equal their scores. For example, if there are three systems, one is significantly better, and the other two are hard to tell, we will assign score 2 to the first system, and 0.5 to each of the rest two systems. We then average the scores of each system over all topics.

Baseline v.s. Zero-order v.s. First-order				
Dataset	#Label	Baseline	Ngram-0-B	Ngram-1
SIGMOD	1	0.76	0.75	**1.49**
SIGMOD	5	0.36	1.15	**1.51**
AP	1	0.97	0.99	**1.02**
AP	5	0.85	0.66	**1.48**

Table 5: Effectiveness of topic labeling
A higher score means that the system tends to be ranked higher.

Basic results: In Table 5, we compare the labels generated using the baseline method (i.e., picking high probability words), 0-order relevance (ngrams, normalized with background probability $p(w|B)$), and 1st-order relevance. For each group of systems, we compare both the top 1 label, and the top 5 labels they generate. From this table, we can make several observations: (1) In all cases, the labels extracted with 1st-order relevance are most preferred by the assessors, indicating that the first-order relevance method is overall the best presumably due to the fact that it can capture the overall topic distribution through context. (2) The preference of first-order relevance labels over the baseline labels is more significant on SIGMOD than on AP. This is likely because phrases are more frequently used and more discriminative in scientific literature than in the news domain. For example, informative phrases such as “mining association rules” are quite common in database literature, whereas common phrases in news articles tend to be general terms such as “united states” and “last year.” This suggests that phrases are generally good labels for scientific topics, but for other genres of text, it may be interesting to explore other candidate labels, such as short sentences. (3) The preference of the first-order relevance labels over the baseline labels is stronger when five labels are considered than when one label is considered. This may be because the preference is amplified when more labels are considered. (4) The labels of 0-order relevance seem to be comparable with those of the baseline except in one case (i.e., 5 labels on SIGMOD) when the 0-order relevance labels are strongly preferred. This again suggests that phrases are not so useful for labeling topics in the news domain, but they are more useful for the literature domain. Overall these results show that the first-order relevance method for automatic labeling of topic models is the best among all the methods.

Noun Phrases v.s. Ngrams			
Dataset	#Label	Chunk-1	Ngram-1
SIGMOD	1	0.40	**0.60**
SIGMOD	5	0.16	**0.83**
AP	1	0.41	**0.59**
AP	5	**0.55**	0.44

Table 6: Ngrams vs. noun phrases as labels

Ngrams vs. noun phrases as candidate labels: To see which of the two methods for generating candidate labels (i.e., ngrams and noun phrases) is better, we compare them on both data sets in Table 6. Interestingly, for the SIGMOD dataset, using statistically significant ngrams as labels is much better than using noun phrases generated by the NLP Chunker, while on the AP dataset the performance of the two types of candidate labels is closer, and in some cases the noun phrases perform even better than ngrams. This may be because the models used by the NLP Chunker are trained on general domains, and not tuned for parsing scientific literature. In general, using significant ngrams appears to be more robust; moreover, this method can also be applied to any genre of texts.

Normalization in 0-order relevance: We now look into the influence of the normalization strategy on the performance of 0-order relevance. In Table 7, we compare 1st-order relevance with 0-order relevance when using two different normalization strategies – uniform normalization (normalization with uniform distribution) and background normalization (normalization with a background distribution). We see that background normalization, although intuitively appealing, does not really help here. The using of background normalization fails to decrease the difference between the 0-order relevance to the better method, the 1-order relevance. Indeed, it even makes the 0-order labels worse when applied on AP. The reason is because the topic models extracted with PLSA are already discriminative due to the use of a background component model (see Section 5.1), thus further normalization with background is not useful, and uniform normalization is actually more robust in this case.

Normalization	Dataset	NGram-0	NGram-1	Diff.
Using Uniform	SIGMOD	0.36	0.64	**0.28**
	AP	0.43	0.57	**0.14**
Using Background	SIGMOD	0.37	0.63	**0.26**
	AP	0.26	0.74	**0.48**

Table 7: Uniform vs. background normailzation for 0-order relevance (5 labels)

To see if background normalization is useful when the extracted topic models are not discriminative (i.e., high probability words are non-informative words), we apply the topic labeling techniques to the topic models extracted with LDA, where neither is a background model included, nor are the stopwords pruned. The results are selectively presented in Table 8.

In Table 8, we show three sample topics extracted with LDA and their corresponding labels generated using 1st-order relevance and 0-order relevance with different normalization methods. Without pruning stopwords or using a background model, we end up having many non-informative words on the top of each topic model; to better illustrate the meaning of each topic, at the bottom part of the topic word distribution, we also present some more discriminative terms. We see that the 1st-order relevance still generates good discriminative labels even though the high probability words of the original topic model are all non-informative words. This is because the 1st-order relevance captures the entire context of the topic model. In contrast, with uniform normalization, the 0-order relevance would be biased to assign high scores to non-informative phrases such as "real data" and "their data" when the top probability terms of the topic model are non-informative (e.g., "the", "their") or too general (e.g., "data", "large"). With normalization by background model $p(w)$, we can penalize a phrase with non-informative or general words, thus alleviate this problem. However, in this way, the top ranked labels tend to be too specific to cover the general meaning of the topic (e.g., "integrity constraints", "transitive closure", etc). Thus overall we see that modeling the semantic relevance with first-order relevance is most robust because the semantics is inferred based on the context of the entire distribution.

Upper bound analysis: How much room is there to further improve the topic labeling method? We can answer this question by looking into how much worse the automatically generated labels are than those generated manually. Thus we ask a human annotator to generate topic labels manually, and ask two different assessors to compare the system-generated labels with the human-generated labels. In Table 9, we see that although the system-generated labels are good, the assessors still consider human-generated labels to be better. This implies that there is still much room to improve the quality of the automatically generated topic labels. Interestingly, the difference between system generated and human generated labels is less significant on the SIGMOD data than on the AP data, suggesting that literature topics may be easier to label than news topics.

System v.s. Human			
Dataset	#Label	Ngram-1	Human
SIGMOD	1	0.35	**0.65**
SIGMOD	5	0.25	**0.75**
AP	1	0.24	**0.76**
AP	5	0.21	**0.79**

Table 9: Comparison with human generated labels

5.3 Labeling Document Clusters

In Section 4.1, we discussed that the topic labeling method could actually be applied to any text information management tasks where a multinomial word distribution is involved. Here we look into one such application – labeling document clusters. In this experiment, we cluster the SIGMOD abstracts with the K-Medoids algorithm [13], and try to utilize the topic labeling method to label the clusters. Specifically, we estimate a multinomial word distribution for each cluster based on its member documents using the maximum likelihood estimator. The proposed topic labeling technique can then be applied on the estimated term distributions, and the top ranked phrases are used to label the original cluster.

Cluster Labels	$\vert d\vert$	**Cluster Medoids (Title)**
multivalued dependencies, functional dependencies	167	A complete axiomatization for functional and multivalued dependencies in database relations
two locking, concurrency control	86	Performance of B-tree concurrency control algorithms
nearest neighbor, similarity search	69	Optimal multi-step k-nearest neighbor search
approximate answering, approximate query	184	Approximate XML query answers

Table 10: Labeling document clusters: K-Medoids

The generated cluster labels, along with the number of documents and the title of the medoid document of each cluster are shown in Table 10. By comparing the cluster labels with the medoid documents, we see that the topic labeling technique is also effective to label document clusters. This experiment shows that the use of topic labeling is not limited to statistical topic modeling; it is potentially applicable to any tasks in which such a multinomial term distribution is involved.

5.4 Context-Sensitive Labeling

In this section, we evaluate the effectiveness of our topic labeling method for *cross-context* labeling/interpretation of topic models. We extract 30 topics from SIGMOD proceedings, but use the phrases extracted from SIGIR abstracts, and KDD abstracts to label the topics. We simulate the scenario in which the system does not know where the topics are extracted, thus it would simply use any context collection "familiar" to the system (i.e., SIGIR or KDD collections in our experiments). The results are presented in Table 11.

The results are interesting. The labels generated from different contexts generally capture the *biased* meaning of

Model	**Labels**	**Model**	**Labels**	**Model**	**Labels**
the, of,	**1st-order:**	the, of	**1st-order:**	the, of,	**1st-order:**
a, and,	foreign key, data integration,	a, to	iceberg cube, data cube,	a, and	clustering algorithm, clustering
in, data,	schema matching, query rewrite,	and, is	data cubes, two types,	to, data	structure, data bubbles, distance
...	web sites, deep web	...	fact table	...	function, very large
constraints	**0-order(u):**	data	**0-order(u):**	clustering	**0-order(u):**
database	data integration, data sources,	cube	data cube, user query,	time	large data, data quality,
integration	real data, their data,	query	large data, data cubes,	clusters	series data, data applications,
sources	data model, database design	system	data structure, over data	databases	high data, clustering algorithm
content	**0-order(b):**	information	**0-order(b):**	large	**0-order(b):**
design	integrity constraints, dynamic	olap	iceberg cube, m se,	performance	transitive closure, subsequence
information	content, sql statements, foreign	multimedia	data cubes, data cube,	quality	matching, data bubbles, clustering
schema	key, schema matching	algorithm	line analytical	algorithm	algorithm, pattern matching

Table 8: Labeling LDA topics: 1st-order relevance is most robust

Topic	**Labels**	**Topic**	**Labels**
tree	**SIGMOD Labels**	views	**SIGMOD Labels**
trees	r tree, b trees,	view	materialized views,
spatial	index structures	materialized	view maintenance, data warehouses
r	**KDD Labels**	maintenance	**KDD Labels**
b	tree algorithm, decision trees	warehouse	decision support
disk	tree construction	tables	business intelligence
dependencies	**SIGMOD Labels**	sampling	**SIGMOD Labels**
functional	multivalued dependencies,	estimation	selectivity estimation
cube	functional dependencies, iceberg cube	approximate	random sampling, approximate answers
multivalued	**SIGIR labels**	histograms	**SIGIR Labels:**
iceberg	term dependency	selectivity	distributed retrieval,
buc	independence assumption	histogram	parameter estimation, mixture models

Table 11: Labeling database topics with different contexts

the topic from the view of that context. For example, the database topic about R-tree and other index structures assigns high probability to words like "tree" and "trees"; the results show that, when interpreted in the data mining context, these high probability words may cause the topic to be labeled with "decision trees" and "tree algorithms", suggesting a different, but related interpretation of "tree." Also, our results suggest that when seeing a topic word distribution with high probability words such as "sampling" and "estimation", database researchers may interpret it as "selectivity estimation" or "approximate answers", while information retrieval researchers interpret it as about "parameter estimation" of "mixture models", which is more relevant to their background.

This experiment shows that the topic labeling technique can be exploited to infer context-sensitive semantics of a topic model through labeling a general topic with different contexts. This provides an alternative way to solve a major task in contextual text mining: extracting general topics and analyzing the variation of their meanings over contexts.

Note that this effect can be only achieved when the first-order semantic relevance is used, since the zero-order relevance is independent of a context.

6. RELATED WORK

To the best of our knowledge, no existing work has formally studied the problem of automatic labeling of multinomial topic models. There has been a large body of work on statistical topic models [11, 4, 28, 22, 9, 2, 3, 16, 18, 19, 14, 24], most of which uses a multinomial word distribution to represent a topic. In some recent work, [23] generalized the representation of a topic model as a multinomial distribution over ngrams. Such topics are labeled with either top words in the distribution or manually selected phrases. The method we proposed can automatically generate meaningful phrase labels for multinomial topic models and can be applied as a post-processing step for all such topic models, to interpret the semantics of these topics models extracted from text data.

As we use phrases as candidate labels, our work is related to phrase extraction, including shallow parsing/chunking in natural language processing (e.g., [15, 10]), and N-gram phrase extraction with statistical approaches (e.g., [7, 26, 1, 6]). A better phrase extraction method could benefit topic labeling as a better preprocessing procedure.

Text summarization aims at extracting/generating sentence summaries for one/multiple documents (e.g., [21]). The summary can be as short as titles [12]. However, no existing work has been done for summarizing a multinomial distribution of words, or a statistical topic model. Since most topic models assume that a document covers multiple topics, it is also difficult to cast topic model labeling as summarizing documents. The topic labeling approach, on the other hand, provides a novel method to label a set of documents.

A major task in contextual text mining is to extract topics and compare their content variations over different contexts [28, 22, 17, 16, 18]. Our proposed topic labeling approach provides an alternative way to infer the context-sensitive semantics of topic models.

7. CONCLUSIONS

Statistical topic modeling has been well studied recently, with applications to many machine learning and text mining tasks. Despite its high impact, however, there is no existing method which could automatically generate interpretable labels capturing the semantics of a multinomial topic model. Without understandable labels, the use of topic models in real world applications is seriously limited. In this paper, we formally study the problem of automatic labeling of multinomial topic models, and propose probabilistic approaches to label multinomial word distributions with meaningful phrases. We cast the labeling problem as an optimization problem involving minimizing Kullback-

Leibler divergence between word distributions and maximizing mutual information between a label and a topic model.

Empirical experiments show that the proposed approach is effective and robust when applied on different genres of text collections to label topics generated using various statistical topic models (e.g., PLSA and LDA). The proposed topic labeling methods can be applied as a post-processing step to label any multinomial distributions in any text context. With reasonable variations, this approach can be applied to any text mining tasks where a multinomial term distribution can be estimated. This includes labeling a cluster of documents and inferring the variation of semantics of a topic over different contexts.

There are many possible extensions to this work. First, there is room to further improve the quality of topic labels, including a potentially better way to select candidate labels. Second, how to incorporate prior knowledge, such as a domain ontology, is also an interesting research direction. Third, it would be interesting to study how to generate labels for hierarchical topic models.

8. ACKNOWLEDGMENTS

This work was in part supported by the National Science Foundation under award numbers 0425852 and 0428472. We thank Jing Jiang for her help on evaluating the labeling effectiveness.

9. REFERENCES

[1] S. Banerjee and T. Pedersen. The design, implementation, and use of the ngram statistics package. pages 370–381, 2003.

[2] D. Blei and J. Lafferty. Correlated topic models. In *NIPS '05: Advances in Neural Information Processing Systems 18*, 2005.

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120, 2006.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[5] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR '98*, pages 335–336, 1998.

[6] J. Chen, J. Yan, B. Zhang, Q. Yang, and Z. Chen. Diverse topic phrase extraction through latent semantic analysis. In *Proceedings of ICDM '06*, pages 834–838, 2006.

[7] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, 1990.

[8] W. B. Croft and J. Lafferty, editors. *Language Modeling and Information Retrieval*. Kluwer Academic Publishers, 2003.

[9] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl.1):5228–5235, 2004.

[10] J. Hammerton, M. Osborne, S. Armstrong, and W. Daelemans. Introduction to special issue on machine learning approaches to shallow parsing. *J. Mach. Learn. Res.*, 2:551–558, 2002.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of ACM SIGIR'99*, pages 50–57, 1999.

[12] R. Jin and A. G. Hauptmann. A new probabilistic model for title generation. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, 2002.

[13] P. J. Kaufman, Leonard; Rousseeuw. *Finding groups in data. an introduction to cluster analysis.* Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics. Wiley. New York., 1990.

[14] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 577–584, 2006.

[15] C. D. Manning and H. Schtze. *Foundations of statistical natural language processing.* MIT Press, Cambridge, MA, USA, 1999.

[16] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of WWW '06*, pages 533–542, 2006.

[17] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceeding of KDD'05*, pages 198–207, 2005.

[18] Q. Mei and C. Zhai. A mixture model for contextual text mining. In *Proceedings of KDD '06*, pages 649–655, 2006.

[19] D. Newman, C. Chemudugunta, and P. Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686, 2006.

[20] P. Pantel and D. Lin. Discovering word senses from text. In *Proceedings of KDD '02*, pages 613–619, 2002.

[21] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Comput. Linguist.*, 28(4):399–408, 2002.

[22] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of KDD'04*, pages 306–315, 2004.

[23] X. Wang and A. McCallum. A note on topical n-grams. In *University of Massachusetts Technical Report UM-CS-2005-071*, 2005.

[24] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of KDD '06*, pages 424–433, 2006.

[25] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of SIGIR '06*, pages 178–185, 2006.

[26] C. Zhai. Fast statistical parsing of noun phrases for document indexing. In *Proceedings of the fifth conference on Applied natural language processing*, pages 312–319, 1997.

[27] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM '01*, pages 403–410, 2001.

[28] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of KDD'04*, pages 743–748, 2004.