

Topic Models and a Revisit of Text-related Applications

Viet Ha-Thuc
Computer Science Department
The University of Iowa
Iowa City, IA 52242, USA
hviet@cs.uiowa.edu

Padmini Srinivasan
School of Library and Information Science and
Department of Management Sciences
The University of Iowa
Iowa City, IA 52242, USA
padmini-srinivasan@uiowa.edu

ABSTRACT

Topic models such as aspect model or LDA have been shown as a promising approach for text modeling. Unlike many previous models that restrict each document to a single topic, topic models support the important idea that each document could be relevant to multiple topics. This makes topic models significantly more expressive in modeling text documents. However, we observe two limitations in topic models. One is that of *scalability* as it is extremely expensive to run the models on large corpora. The other limitation is the inability to model the key concept of *relevance*. This prevents the models from being directly applied to goals such as text classification and relevance feedback for query modification; in these goals, items relevant to topics (classes and queries) are provided upfront. The first aim of this paper is to sketch solutions for these limitations. To alleviate the scalability problem, we introduce a one-scan topic model requiring only a single pass over a corpus for inference. To overcome the latter, we propose relevance-based topic models that have the advantages of previous models while taking the concept of relevance into account. The second aim, based on the proposed models, is to revisit a wide range of well-known but still open text-related tasks, and outline our vision on how the approaches for the tasks could be improved by topic models.

Categories and Subject Descriptors

H.1.1 [Models and Principles]: General

General Terms

Algorithms, Performance, Design, Experimentation, Languages, Theory

Keywords

Topic models, LDA, relevance-based language models, Gibbs sampling

1. INTRODUCTION

Topic models, including aspect model (pLSI) [10] and its enhanced version Latent Dirichlet Allocation (LDA) [3, 19] have been accepted as a promising approach for text modeling [21]. The strength of topic models is their strong theoretical framework supporting the idea that each document is a mixture of multiple

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PIKM'08, October 30, 2008, Napa Valley, California, USA.
Copyright 2008 ACM 978-1-60558-257-3 /08/10...\$5.00.

topics, where topics are multinomial distributions over words. This allows topic models to consider the different themes mentioned in a document and overcome the strict restriction of previous models for text modeling that assume each document is related to exactly one topic as in cluster model [13], Robertson & Sparck-Jones probabilistic model [17], or relevance-based language models [12].

Because of the above characteristic, topic models also provide an efficient representation for documents or user queries via topic mixing proportions. Compared to the traditional vector space representation with weights such as *tf-idf* on terms (words or stems), topic mixing representation is significantly shorter (the number of terms vs the number of topics) while still preserving essential statistical relationships [3]. Even more, with some further extension, as we will show later on in this study, topic models could be used for modeling the *relevance* of predefined topics, which is key in a wide class of applications such as text classification, topic-specific keyword finding, topic relationship mining, and relevance feedback for information retrieval.

This research* begins by identifying two key limitations in current topic models. The first limitation pertains to scalability which is indicated by the fact that most previous works use topic models for *representing documents in medium-size corpora* (less than 2GB) [3, 5, 7, 10, 19, 21, 23, 24]. The challenge is that the inference algorithms for the models require many scans over a corpus. When the corpus cannot fit into internal memory, the algorithms require many external memory operations (e.g. disk accesses), which are extremely costly, per scan. This limits the scalability of topic models with increasing corpus size.

A second limitation of topic models affects the range of suitable applications; namely these models do not explicitly take the concept of relevance into account. Given a corpus and a number K , topic models operate by discovering K topics in the corpus. (Note that a different iteration of the topic models inference algorithm may give rise to a different set of K topics). As the resulting topics are synthetic; they do not explicitly correspond to the prior knowledge of human beings regarding topics in the corpus domain. Thus these models are not directly applicable to goals such as text classification, keyword finding, and relevance feedback for query modification, where specific topics (classes and queries) and example relevant items are provided upfront.

Having identified the two limitations the *first goal of this paper* is to sketch solutions. For scalability, we propose the idea of one-scan models that require only single pass through a corpus for inference. A sampling technique is used to discover K topics in the

* This research will be the basis of the first author's doctoral research proposal.

corpus. Then, the corpus is loaded to internal memory, (in suitably-sized) chunk by chunk. For each chunk, using the K topics discovered previously, we compute topic mixing proportions for the documents. After that, the memory is emptied for the next chunk of documents. A similar idea has successfully been explored for clustering very large datasets in previous research including our own (Bradley et al. [4], Farnstrom et al. [6], Ha-Thuc et al. [8]). So, we expect a similar success in topic models.

To extend the application range to the class of tasks where we need to model relevance, we introduce relevance-based topic models. Relevance-based topic models have the advantage of previous topic models in that in addition to allowing multiple topics per document, they explicitly consider relevance as in the relevance model of Robertson & Sparck-Jones [17] and relevance-based language models [12]. Specifically, our models assume that each document d in the relevant set of a topic of interest t is generated by a mixture of three topics: topic t itself, a background topic, b , which captures common words, and topic $to(d)$ (other themes in the document) capturing non-relevant parts (noise) in the document. Because we model the background and $to(d)$ topics, general stop words and domain-specific stop words as well as the noisy portions in the relevant document d are automatically identified. These are done with respect to the particular topic t for which d is relevant. Thus only the parts of d generated by t (the really relevant parts) contribute to the resulting multinomial distribution, explicitly associated with topic t .

The *second goal of this paper* is to revisit a wide range of text-related tasks including single-label and multiple-label text classification, relevance feedback, word stemming, and to indicate the potential of the topic models in solving the problems compared to their current approaches.

The rest of paper is organized as follows. In Section 2, we review the state-of-the-art topic model, LDA, we then introduce our one-scan LDA to alleviate the issue of scalability. In Section 3, we propose relevance-based topic models appropriate for a wide range of applications where we need to model the concept of relevance. Section 4 outlines the approaches based on the proposed models for well-known text-related problems. Section 5 presents some initial experimental results. Finally, Section 6 is our conclusions.

2. LDA AND ONE-SCAN MODEL

2.1 LDA from the Literature

LDA is a generative probabilistic model of a corpus. It describes how documents in the corpus are generated:

- 1) For each topic z in $\{1 \dots K\}$, where K is the number of topics:
 - Pick a multinomial distribution Φ_z from a W -dimensional Dirichlet distribution $Dir(\beta)$, where W is the number of words in the vocabulary set.
- 2) For each document d in the corpus:
 - a) Pick a multinomial distribution θ_d from K -dimensional $Dir(\alpha)$.
 - b) For each token in the document d :
 - i) Pick a topic z in $\{1 \dots K\}$ from θ_d
 - ii) Pick a word w in the vocabulary set from the multinomial distribution Φ_z

This generative process is illustrated by the graphical model using the plate notation in Fig. 1, where $N_d, |D|$ are the number of tokens in document d and the number of documents in the corpus respectively; the numbers at the right low corner of plates (i.e. $K, N_d, |D|$) indicate the number of iterations of the corresponding plates; α and β are hyper parameters of Dirichlet distributions, which are often treated as predefined constants [7, 9]; w 's are tokens observed in the corpus.

Given a corpus and a value of K , LDA infers the latent variables z, Φ, θ . We could use Gibbs sampling for inference. The inference algorithms for LDA are described in detail in [1, 3, 19].

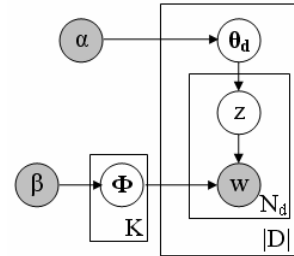


Figure 1: Graphical model representation of LDA

So, LDA discovers K topics (Φ_z) present in the corpus, and assumes each document d in the corpus is generated by a mixture (θ_d) of the K topics. Thus, the topic mixture provides an explicit way to represent documents. Compared to *tf-idf* representation in vector space model, latent topic representation is not only significantly shorter but also utilizes more inter- and intra-document statistics [3].

2.2 One-Scan Model

A key challenge with LDA is scalability. The inference algorithms for LDA need to access every data element (i.e. token) in every iteration. When the corpus cannot fit into internal memory, the algorithms require many external memory operations (e.g. disk accesses), which are extremely costly, per iteration. Typically, the Gibbs sampling-based inference algorithms for LDA take from hundreds to thousands of iterations to converge [19, 21, 24]. That makes topic models not well scalable with corpus sizes.

To make LDA applicable to realistic corpora, we must minimize the number of external scans. Ideally, in terms of scalability, the model should require only one scan for inference. We introduce such a model in Fig. 2.

The inference process includes two phases. First, a random subset that fits into memory is used to approximately infer K topics. Therefore, we do not need to scan through the corpus for this phase. Second, given the K topics, the model infers topic mixture for each document. It is worth noting that the topics are completely known at this phase, so the topics are modeled as observed variables (shaded circles in Fig. 3) and their distributions stay unchanged during the phase. The variables needed to be inferred include latent topic z of each token and topic mixing proportion θ_d for each document d . The latent variable z is sampled from its posterior probability given its Markov blanket (See (2) in Fig. 2). The topic mixing variable θ_d is estimated by formula (1) in Fig. 2, where $n^{(t-1)}_{i,j}$ is the number of times in sample $(t-1)$ that topic j is assigned to some token in document i . The final value of θ_d could be computed by averaging over multiple samples. Then, we fill the buffer with next chunk of documents (Step 2.4). So, we need

exactly one external scan to compute topic mixing proportion for all documents in the corpus no matter how large the corpus is in comparison to internal memory size.

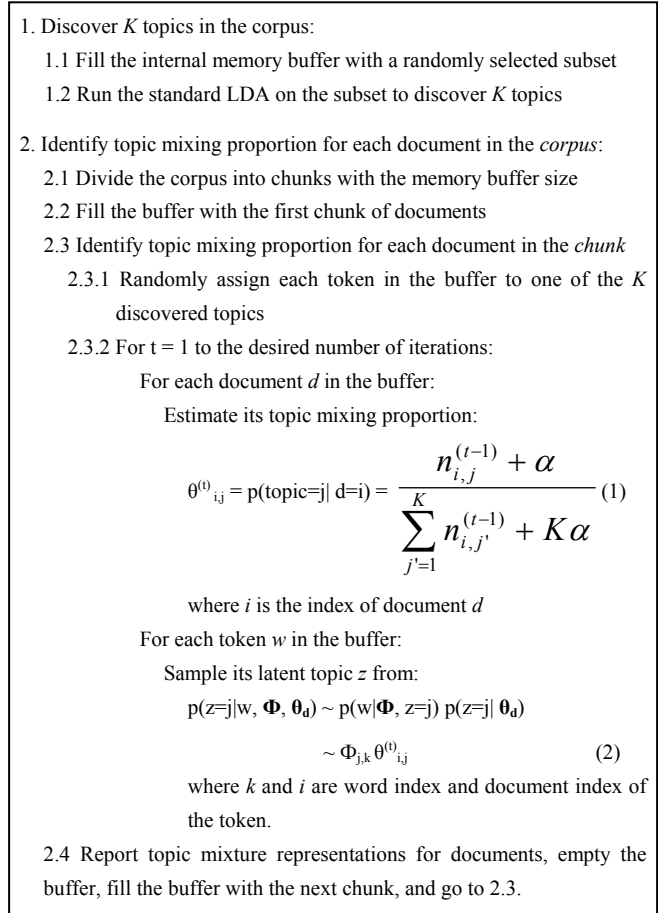


Figure 2 – One-scan LDA

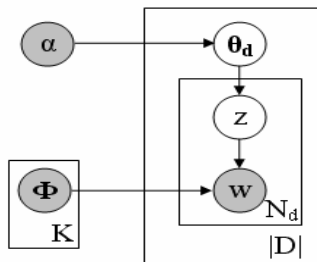


Figure 3 - One-scan LDA at the second phase

There are some alternatives for the first phase in the model above. For instance, we could load the corpus, chunk by chunk, as in the second phase, and semantically compress [8, 4] the chunks. After that, we run the standard LDA on the compressed set to discover K topics. The approach needs an additional external scan but takes the content of the whole dataset into account instead of only sampled documents. We plan to do experiments on both approaches and analyze the results of these approaches in comparison with the standard LDA.

3. RELEVANCE-BASED TOPIC MODELS

As described above, LDA does not explicitly model the concept of *relevance*, which is key in numerous applications [9, 18]. Consequently, there is no explicit mapping between the resulting topics generated by LDA and the topics of interest to a user or user community. Therefore, the approach could not be applied directly for applications, such as text classification and relevance feedback for query modification, where topics (classes and queries) and example relevant items are provided upfront. On the other hand, explicit relevance models popularly used in information retrieval community, such as Robertson & Sparck-Jones probabilistic model [17] and relevance-based language models [12] often make a strict assumption that if a document is relevant to a topic, the whole document is relevant to that topic. This assumption is, nonetheless, not true in many practical cases where only a part of the document is actually relevant to the topic.

This study proposes relevance-based probabilistic topic models, an extension of LDA, to bridge the two separate approaches mentioned above. The models assume that each document d in the relevant set of a topic of interest t is generated by a mixture of three topics: topic t itself, a background topic, b , which captures words that are common in general or common in the particular domain, and $to(d)$ (other themes in the document) capturing non-relevant parts in the document. Only the parts generated by t (the really relevant parts) contribute to the resulting distribution over words for topic t . In the models, the topics are predefined by users and are explicitly associated with the resulting distributions. In the other words, relevance is explicitly modeled. On the other hand, the models relax the assumption of relevance-based language models. More specifically, the models take advantage of LDA's multiple topic framework to support the important fact that a document relevant to a topic t might also talk about some other themes rather than t alone.

The contributions of topics in documents are automatically determined by intra- and inter-document statistics. Intuitively, the words frequently appearing in the whole corpus are likely generated by the background topic, the words frequently appearing in the relevant set of a topic of interest t but not the whole corpus are likely generated by this topic, and the words appearing in only a particular document d in the relevant set of topic t but neither frequently in the other documents of this relevant set nor in the background are likely generated by $to(d)$.

In this study, we introduce two kinds of relevance-based probabilistic topic models: batch topic model and online topic model. The first is applicable for some tasks where we would like to model K_0 topics simultaneously, e.g. text classification or batch search. In this model, the background topic covers the words that are common across all topics, while the distribution of each topic of interest concentrates on features that discriminate the current topic from the other $(K_0 - 1)$ topics. The second could be applied for tasks where we have only one topic at a time, e.g. online information retrieval. In this case, the background topic captures the language use in the whole corpus so that the distribution of the current topic would not waste its probability mass on common features, instead it could focus more of the unique features on the topic. Due to this distinction, the two models use slightly different methods to infer the background topic.

3.1 Batch Topic Model

Batch topic model is a relevance-based probabilistic model describing the process of generating relevant documents of a set of topics as follows.

- 1) Pick a multinomial distribution Φ_b for the background topic(b) from a W -dimensional Dirichlet distribution $Dir(\beta)$.
- 2) For each topic t in K_0 topics of interest:
 - a) Pick a multinomial distribution Φ_t for t from the W -dimensional $Dir(\beta)$.
 - b) For each document d relevant to t :
 - i) Pick a multinomial distribution $\Phi_{t_o(d)}$ for the topic covering themes other than t that are also mentioned in d from the W -dimensional $Dir(\beta)$.
 - ii) Pick a multinomial distribution θ_d from a 3-dimensional $Dir(\alpha)$, each element of θ_d corresponds to a topic in $x_d = \{b, t, t_o(d)\}$.
 - iii) For each token in document d :
 - (1) Pick a topic z among the three topics in x_d from multinomial θ_d .
 - (2) Then, pick a word from the corresponding multinomial distribution Φ_z .

This process is described by the graphical model using plate notation in Fig. 4. In the Figure, w and x_d are observable variables and denoted by shaded circles; z , θ and Φ are hidden variables and denoted by un-shaded circles; α , β are parameters of Dirichlet distributions.

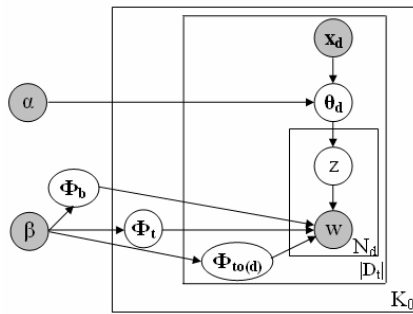


Figure 4: Batch topic model

Given K_0 topics and their relevant sets (i.e. sets of relevant documents), we could use Gibbs sampling technique, as in the standard LDA, to infer latent topic z , topic word-distributions Φ , and topic mixing proportion θ . It is worth noting that in the case some of the K_0 topics are conceptually overlapping, a document could appear in several relevant sets. In such case, the document has a corresponding number of copies in the corpus. However, each copy plays a different role contributing different parts to the topic to which it belongs. For instance, a paper could be relevant to two topics “*machine learning*” (ML) and “*information retrieval*” (IR) (Fig. 5 – for simplicity, we ignore the background part in this example), so it appears in the relevant sets of both ML and IR. In the first copy, parts (1) and (2) are relevant while part (3) is non-relevant. In the second copy, parts (2) and (3) are relevant while part (1) is non-relevant.

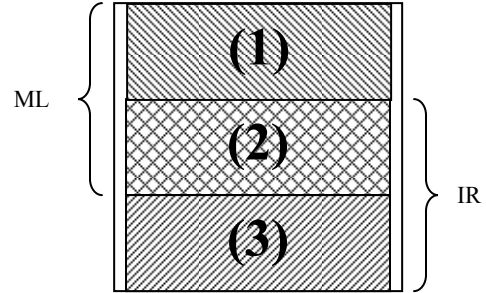


Figure 5 – A document relevant to both topics ML and IR

3.2 Online Topic Model

Online topic model is a relevance-based probabilistic model describing the process of generating the relevant set of a given topic t . Similar to the batch topic model described in previous section, the online topic model assumes that each document d in the relevant set is generated by three topics: the background b , the topic of interest t , and $t_o(d)$ which rolls up every other theme but t , the topic for which the document d is relevant. The notion of the background topic, however, is slightly different.

In the batch topic model, the background covers common word features of all given topics so that each of the topics could spend its probability mass on its discriminative features that distinguish itself from the rest of the topics. The purpose of the background topic is, therefore, increasing the margins among the distributions of the topics of interest. So, the background topic should be dependent on the given topics and modeled as a latent variable (Fig. 4). In online topic model, the background topic, b , represents the common language used in the whole corpus. So, it should be independent of the topic and could be estimated in advance by term frequencies in the corpus. Therefore, it is modeled as an observed variable (Fig. 6).

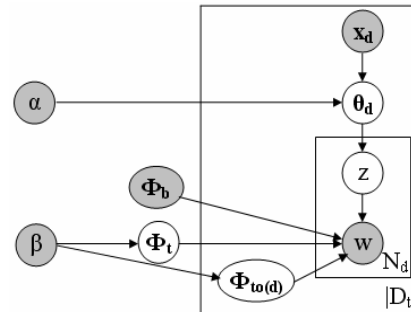


Figure 6: Online topic model

4. APPLICATIONS

In this section, we revisit four well-known but still open text-related tasks. We outline our vision on how topic models such as LDA, one-scan LDA, relevance-based online topic or relevance-based batch topic models could solve these tasks, and potentials of topic models-based approach in comparison to current approaches.

4.1 Single-Label Text Classification

Single label text classification is the task of classifying each document into exactly one of K_0 given topics (classes). The common approach for this task is using relevant sets of the topics to train a classifier, and then using the trained classifier to classify unseen documents. Previous methods often assume that every part

of a positive training document of a topic is relevant to that topic. However, in practice, it is often the case that only one or a few parts in the training document are really relevant to the topic.

To overcome this limitation, we plan to apply relevance-based batch topic model for single-label text classification. Given relevant sets of K_0 topics, we use the model to estimate $\Phi_{j,k} = p(\text{word} = k | \text{topic} = j)$ ($1 \leq j \leq K_0$, $1 \leq k \leq W$). We then can use Bayes formula to estimate the posterior probability $p(\text{topic} | \text{doc} = w_1 w_2 \dots w_N)$. Recall that in our model, the really relevant parts in each document are automatically determined by their statistical correlation to the rest of the positive training set. Only these parts contribute to the final results $\Phi_{j,k}$. Therefore, the approach is robust to “impurities” in the training sets.

We also plan to apply the multi-topic model for text classification without any human-labeled training data. Instead, we will use as training sets of documents returned from a global search engine (e.g. Google) or an intranet search engine, retrieved by the topics themselves. The challenge of this approach is that there is a lot of noise in the returned sets. The ability to automatically detect non-relevant parts in documents of our model is key to tackle this challenge.

4.2 Multi-Label Text Classification

Another natural direction is to use the batch topic model for multi-label text classification, where a document could be relevant to several topics (classes) [14]. For instance, a research paper about learning to rank may have some parts of the paper relevant to machine learning (ML) only, some parts relevant to information retrieval (IR) only, and some others relevant to both of the topics (See Fig. 5).

McCallum [14] proposes an approach that assumes each word in a document is *exclusively* generated by one of the topics to which the document is relevant or a background topic. So, his approach cannot model the overlap among the topics (e.g. ML and IR in our example). This is because of the difference in nature of two concepts: generation and relevance. Generation is exclusive (in his approach) whereas relevance is by nature inclusive. Another popular approach for multi-label multi-class text classification problem is building a binary classifier (relevance and non relevance) for each class [11]. The approach again assumes all parts of the document are relevant to both ML and IR. This assumption, as mentioned before, does not hold.

Our multi-topic model, as described in Section 3.1, could overcome limitations of both approaches since it is able to model the overlap among topics as well as automatically detect non-relevant parts in training documents.

4.3 Relevance feedback

The third planned direction of future research is to apply the online topic and batch topic models for relevance feedback. Given K_0 queries (batch search) and their feedback documents, we use batch topic model to estimate $\Phi_{j,k} = p(\text{word} = k | \text{topic} = j)$ ($1 \leq j \leq K_0$, $1 \leq k \leq W$), and rank the keywords w.r.t topics by the posterior probability. Then queries could be expanded by adding the top relevant keywords on them. In online search, we have only query at each time, we could use online topic model to estimate $\Phi_{j,k}$. Finally, we submit the expanded queries to a search engine.

Tan *et al.* [20] point out that the most challenging issue in finding the keywords from feedback documents is that non-relevant terms, occurring along with relevant ones in the documents, could cause undesired effects. This is exactly one of the problems that our models are designed to solve. We remind the reader that each document is considered to be a mixture of three topics (b , t and $to(d)$). We plan to use TREC datasets for these experiments and investigate the effectiveness of extracted feedback terms.

4.4 Word Stemming

Stemming is a technique to reduce variant word forms into common morphological roots [22]. For instance, *explains*, *explained*, *explaining*, *explanation*, *explainer*, *explainers* are stemmed to the root *explain*. This technique potentially improves the ability to reduce the number of word features (therefore, search space), to match query and document words in information retrieval, and to lead to a better generalization in text classification. Despite these potentials, the literature has not shown agreement on whether stemming is helpful or not. In some cases, stemming helps to improve performances, in some cases it hurts [2]. The reason for the later is that stemmers sometimes over stem several unrelated words into the same group [2], e.g. *race*, *racer*, and *racist*.

To overcome the over stemming problem, Xu and Croft [22] first use a traditional stemmer (e.g. Porter) to create initial equivalence classes containing words stemmed to the same morphological roots. Second, they use the word co-occurrence relation to split each class into subclasses. The subclasses are believed to contain morphologically and semantically related words, thereby over-stemming is reduced. In recent work [2] (2007), Bhamidipati and Pal propose a method for the second phase by taking advantage of category labels of documents. For each word, they estimate the multinomial distribution over categories of documents in which the word appears. Then, they use the distributions to compute the similarity between every pair of words in the initial equivalence classes. After that, the similarities are used to partition each initial class into subclasses.

However, the category information is not available in most realistic datasets. Second, the method assumes every word in a document is relevant to the document’s category. Note that, Bhamidipati et al. consider every token in a document as belonging to the same class. As we mentioned before, this assumption is unrealistic. Third, the method works at the word level, whereas the ambiguity, one of the main reasons of over-stemming errors, happens at the token level. For instance, if the word *race* appears in a document about car sport, it should be grouped with $\{\textit{racing}, \textit{racer}\}$, while if it appears in a document about social science, its likely group is $\{\textit{racist}, \textit{racism}\}$. Similarly, the word *book* could be grouped with $\{\textit{books}\}$ or $\{\textit{booking}, \textit{booked}\}$ depending on the context in which it appears.

Motivated by these limitations, we plan to apply the standard LDA or one-scan model (in the case the corpus is larger than internal memory) for word stemming task. The rationale of this approach is three fold. First, LDA or one-scan model could automatically discover topics in the corpus. Second, different parts of a document could be relevant to different topics. Third, LDA or one-scan model estimates a multinomial over the discovered topics for each *token* in documents (formula (2) – Step 2.3.2, Fig. 2) by taking into account the context (document) in which the token appears, as well as the occurrences of the same word in the other

documents. The distribution reveals the meaning of the token and helps to assign the token to the appropriate stemming subclass.

5. INITIAL EMPIRICAL RESULTS

5.1 An Illustration

In this section, we illustrate some important characteristics of the proposed relevance-based topic models by applying it to a very small and pseudo dataset. The small size of the dataset makes it easy and intuitive to analyze the returned results and illustrate model characteristics. More comprehensive experiments are presented in Section 5.2.

The dataset contains the relevant sets for two topics *machine learning* (ML) and *data mining* (DM) (Table 1). Each document comprises of three parts: general terms (e.g. “*artificial*” or “*intelligence*”), topical terms (e.g. “*learning*” respect to ML or “*mining*” respect to DM), and non-relevant terms (e.g. “*NLP*” in doc 1). Although the dataset is pseudo, it intuitively makes sense. One can imagine that doc 1 is a paper about applications of machine learning in natural language processing, or doc 5 is a data mining paper in biomedical domain.

Table 1 - A pseudo dataset

doc#	class	Document content
1	ML	Artificial intelligence machine learning machine learning training NLP NLP NLP
2	ML	Artificial intelligence machine learning machine learning training speech speech speech speech recognition recognition recognition waves waves waves
3	DM	Intelligence data mining data mining classification clustering
4	DM	Artificial intelligence data mining data mining time series
5	DM	Artificial intelligence data mining data mining classification clustering biomed biomed

We apply batch topic model on the dataset described above. Table 2 shows top words ranked by probability $p(\text{word} | \text{topic})$ of some topics. First we note that *intelligence* and *artificial* occur frequently across the two relevant sets and so they are identified as domain based stop words and appear in the background distribution with high weights. As regards topic representative words, in topic ML, *machine*, *learning* and *training* appear frequently in all relevant documents of this topic but not in other documents (see Table 1). So as expected they are identified by our model as representative of the topic and have dominant roles in the ML distribution (Table 2). Similarly, *mining*, *data* and *cluster* are frequent in the DM relevant set but not in other documents; thus they have high weights in the DM distribution. Notice also that these topical terms have very low weights in the background distribution (Table 2).

More interesting observations may be made by looking at the lower row in Table 2. We see for example, that *biomed* is a highly weighted term for $to(d_3)$. This is because of two reasons. One is that it as it appears in very few documents of the collection (1 out of 5) so it is not considered as part of the background. Second as it appears in very few documents of the training set (1 out of 3 for DM) it is not viewed as topical either. However, it does appear

with high frequency in doc 5, therefore the model learns that there is some other topic (non relevant to DM, in the relative sense) that likely generates this word in this document. Hence, it has a high weight in the $to(d_5)$ distribution. Notice here that *biomed* has the same token frequency in the DM relevant set as *cluster* which was regarded as topical for DM. However their occurrences across the DM training set differ thus accounting for the difference in how the model treats the two words. Consequently also $p(\text{“cluster”} | DM)$ is larger than $p(\text{“biomed”} | DM)$ (0.12 versus 0.02).

We also applied the online topic model sequentially on each topic of the dataset. In this case the background topic is estimated by term frequencies in the whole dataset. The resulting topic-word distributions are quite similar to results above.

Table 2 - Topic-word distributions

ML		DM		Background	
learning	.32	mining	.31	intellig	.34
machine	.31	data	.30	artificial	.30
train	.16	cluster	.12	machine	.06
intellig	.06	classifi	.11	learn	.05
artificial	.05	intellig	.08	data	.05
speech	.03	artificial	.04	mining	.05
recogni	.03	biomed	.02	train	.03
wave	.02	time	.01	classify	.03
nlp	.02	series	.01	cluster	.03

$to(d_1)$		$to(d_2)$		$to(d_3)$	
NLP	.82	speech	.38	biomed	.75
machine	.07	wave	.28	classifi	.10
learn	.06	recogni	.28	cluster	.09
train	.02	machine	.02	artificial	.03
intellig	.01	learn	.02	mining	.01
artificial	.01	train	.01	data	.01
speech	5e-7	intellig	.004	intellig	.01
recogni	5e-7	artificial	.003	recogni	4e-7
wave	5e-7	nlp	1e-7	wave	4e-7

This small example illustrates how the relevance-based topic models are able to correctly distinguish between the different roles of the words across the topics.

5.2 Finding Keywords

This section presents an application of the relevance-based online topic model and batch topic models: finding keywords w.r.t given topics (queries). The application is meaningful because from the users’ point of view, the keywords are crucial in interpreting the meaning of the topic or in clarifying information needs, and from the systems’ point of view, finding descriptive keywords (i.e. feedback terms) is a key step in query expansion [20].

Datasets: We use three datasets. The first one is ML Cora collection of machine learning abstract papers from the Cora corpus [15]. Each of the papers is categorized into seven topics in machine learning. The second one is News5 dataset, a subset of 20 Newsgroup dataset, containing the five *comp.** classes [16] that are somewhat related to each other. The third one is a subset of Reuters-21578 [14] containing documents that are relevant to one of the five most popular topics in Reuters-21578. In order to avoid too short and meaningless documents (e.g. containing only an

acknowledgment), we remove the documents with less than 100 words (including the stop words) out of the datasets. The resulting size of each dataset is around 3000. Then, we apply stop word removal [25] and stemming (Porter). We use class label of each document as the topic of interest to which the document belong.

For judgment, given each topic we use all relevant documents available for training and rank the keywords by the standard *tf-idf*, then manually judge the top 50 keywords. Generally, each topic has about 5-15 relevant keywords. Those are used as gold standards

Methodology: We run the experiment in both batch mode and online mode. In the batch mode, we use the batch topic model to find the keywords for K_0 topics simultaneously (K_0 equals to 7, 5 and 5 in the datasets ML Cora, News5 and Reuters, respectively). In the online mode, we use the single-topic model to find keywords for one topic at each time. Keywords are ranked by the conditional probability $p(\text{word}|\text{topic})$, returned by the batch topic model or the online topic model.

For comparison purpose, we use *Rocchio*, a well known method for relevance feedback as the baseline.

$$Rocchio(w|t) = \alpha * tf-idf(w|R_t) - \beta * tf-idf(w|N_t)$$

where $tf-idf(w|R_t)$ and $tf-idf(w|N_t)$ are *tf-idf* values of word w in relevant documents and non-relevant documents of t , respectively. We fix the value of α equal to 1.0, and tune the different values including 0.05, 0.1, 0.25, 0.5 and 1.0 for β . Then select the best performance as the baseline result for each run. We call this the tuned Rocchio.

We vary the number of the relevant sets per topic for training (10, 25, 50 and 100), and use the gold standards described above to compute MAP (mean average precision) for each method. Given a size, there are many choices of the relevant set for each topic, so we run each method 50 times with different choices, and compute the mean and the two-tailed *p-values* by paired *t-test* comparing one of our methods (the batch topic model or the online topic model) versus the baseline.

Results: The results are shown in Table 3. Note that the cell contains the asterisk symbol (*) means the difference between the corresponding method and the baseline is statistically significant (i.e. $p\text{-value} < 0.05$). As shown in Table 3, the batch topic model is significantly better than the tuned Rocchio in 11/12 cases. Among those, there are 10 cases where the improvement is more than 10% (See Tables 3). Compared to the tuned Rocchio, the online topic model is significantly better in 8/12 cases and worse in 3/12 cases.

Among the datasets, the performances of the three methods on Cora and Reuters are significantly better than on News5. This is perhaps because the topics of Cora and Reuters are more well-defined than topics of News5. Between our two models, the batch topic model working in batch mode and the online topic model working in online model, the first is significantly better in most of the 12 cases. The difference makes sense since the batch topic model uses more information (relevant sets of all topics given in the problem) than online topic does (only relevant set of the current topic along with the whole corpus). More specifically, in the online topic model, the background topic is roughly estimated by term frequencies in corpus (e.g. ML Cora, News5 or Reuters in our experiments), while the batch topic model dynamically tunes the background topic so that the topic best covers the common features among the topics.

Table 3 - MAP

# of rel docs	10	25	50	100
Batch topic model	.263 * (+24.2%)	.391 * (+11.5%)	.502 * (+3.7%)	.603 * (-2.1%)
Online topic model	.242 * (+14.2%)	.352 (+0.4%)	.421 * (-13.1%)	.462 * (-25.1%)
Tuned Rocchio	.212	.351	.484	.616

3(a) - ML Cora dataset

# of rel docs	10	25	50	100
Batch topic model	.160 * (+36.7%)	.235 * (+23.8%)	.350 * (+19.9%)	.471 * (+14.8%)
Online topic model	.109 * (-7.4%)	.225 * (+18.7%)	.360 * (+23.2%)	.448 * (+9.2%)
Tuned Rocchio	.117	.190	.292	.411

3(b) - News5 dataset

# of rel docs	10	25	50	100
Batch topic model	.339 * (+42.3%)	.484 * (+26.4%)	.571 * (+18.8%)	.654 * (+12.5%)
Online topic model	.262 * (+9.86%)	.447 * (+16.7%)	.549 * (+14.2%)	.607 * (+4.43%)
Tuned Rocchio	.238	.383	.481	.581

3(c) - Reuters

6. CONCLUSIONS AND FUTURE WORK

This paper presents a preliminary doctoral dissertation proposal of the first author. In the first part of the paper, we indicate promising potential of topic models in text modeling and observe their current limitations regarding *scalability* and the *inability to model relevance*. In the second part, we propose solutions for these issues by introducing one-scan LDA and relevance-based topic models. One-scan LDA requires only a single external disk scan for inference. That makes the model scalable well with large corpora. Relevance-based topic models have advantages of both traditional relevance-based language models and LDA. Our models explicitly model the relevance. The models also inherit the multiple topic document theoretical framework of LDA to extract background terms as well non-relevant terms in relevant documents. After proposing solutions to the limitations, in the third part of the paper, we revisit a wide range of text-related tasks. We show our vision on how topic models could solve the tasks and potential advantages of topic models-based approaches in comparison to current approaches of each of these tasks.

We have implemented relevance-based topic models including batch topic model as well as online topic model and conducted several initial experiments. The results in Section 5 demonstrate rationales and potential of the models. For future work, we plan to implement one-scan LDA with several alternatives described in Section 2.2. Then, we implement topic models-based approaches for applications described in Section 4, and compare with current methods for each application.

Since the first author is in early stage of his PhD program, any comments either about proposed formal models, sketched

approaches for the four applications, experiment designs, or new directions to apply the models are absolutely welcome. The comments would be very valuable to revise this study for his doctoral dissertation proposal.

7. REFERENCES

- [1] Adrieu, C., Freitas, N., Doucet, A., Jordan, M., *An Introduction to Markov Chain Monte Carlo for Machine Learning*, Machine Learning, 50, 2003.
- [2] Bhamidipati, N., Pal, S., *Stemming via Distribution-based Word Segregation for Classification and Retrieval*, In IEEE Transactions on Systems, Man, and Cybernetics, 37(2), 2007.
- [3] Blei, M., Ng, A., Jordan, M., *Latent Dirichlet Allocation*, Journal of Machine Learning Research, 3, 2003.
- [4] Bradley P.S., Fayyad, U., Reina, C., *Scaling Clustering Algorithms to Large Databases*, In Proceedings of the 4th ACM SIG International Conference on Knowledge Discovery and Data Mining Conference (KDD), 1998.
- [5] Erosheva, E., Fienberg, S., Lafferty, J., *Mixed-membership Models of Scientific Publication*, In Proceedings of National Academy of Science (PNAS), 2004.
- [6] Farnstrom, F. Lewis, J., Elkan, C., *Scalability for Clustering Algorithms Revisited*, In Proceedings of the 6th ACM SIG International Conference on Knowledge Discovery and Data Mining Conference (KDD), 2000.
- [7] Griffiths, T., Steyvers, M., *Finding Scientific Topics*, In Proceedings of National Academy of Science (PNAS), 2004.
- [8] Ha-Thuc, V., Nguyen, D.C., Srinivasan, P., *A Quality-Threshold Data Summarization Algorithm*, In Proceedings of the 6th IEEE International Conference on Research, Innovation and Vision for the Future (RIVF), 2008.
- [9] Hiemstra, D., Robertson, S., Zaragoza, H., *Parsimonious Language Models for Information Retrieval*, In Proceedings of the 27th ACM SIG International Conference on Research and Development in Information Retrieval (SIGIR), 2004.
- [10] Hofmann, T., *Probabilistic Latent Semantic Indexing*, In Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI), 1999.
- [11] Lauser, B., Hotho, A., *Automatic Multi-label Subject Indexing in a Multi-lingual Environment*, In Proceedings of the 7th European Conference in Research and Advanced Technology for Digital Libraries (ECDL), 2003.
- [12] Lavrenko, V., Croft W. B., *Relevance-based Language Models*, In Proceedings of the 24th ACM SIG International Conference on Research and Development in Information Retrieval (SIGIR), 2001.
- [13] Liu, X., Croft, B., *Cluster-based Retrieval Using language Models*, In Proceedings of the 27th ACM SIG International Conference on Research and Development in Information Retrieval (SIGIR), 2004.
- [14] McCallum, A., *Multi-Label Text Classification with a Mixture Model Trained by EM*, In Proceedings of AAAI Workshop on Text Learning, 1999.
- [15] McCallum, A., Nigam, K., Rennie, J., Seymore, K., *Automating the Construction of Internet Portal with Machine Learning*, Information Retrieval, 3, 2000.
- [16] Nigam, K., Ghani, R., *Analyzing the Effectiveness and Applicability of Co-training*, In Proceedings of the 9th ACM Conference on Knowledge and Information Management (CKIM), 2000.
- [17] Robertson, S., Sparck-Jones, K., *Relevance Weighting of Search Terms*, Journal of American Society for Information Science, 27, 1988.
- [18] Sparck-Jones, A., Robertson, S., Hiemstra, D., Zaragoza, H., *Language Modelling and Relevance*, In Croft, B., and Lafferty, J. (eds.) Language Modeling for Information Retrieval, Kluwer Academics, 2003.
- [19] Steyvers, M., Griffiths, T., *Probabilistic Topic Models*, In Landauer *et al.* (eds.) Latent Semantic Analysis: A Road to Meaning, Laurence Erlbaum, 2006.
- [20] Tan, B. et al., *Term feedback for Information Retrieval with Language Models*, In Proceedings of the 30th ACM SIG International Conference on Research and Development in Information Retrieval (SIGIR), 2007.
- [21] Wei, X., Croft, B., *LDA-based Document Models for Ad-hoc Retrieval*, In Proceedings of the 29th ACM SIG International Conference on Research and Development in Information Retrieval (SIGIR), 2006.
- [22] Xu, J., Croft, B., *Corpus-based Stemming Using Cooccurrence of Word Variants*, In ACM Transactions on Information Systems, 16(1), 1998.
- [23] Zhou, D., Ji, X., Zha, H., Giles, L., *Topic Evolution and Social Interactions: How Authors Effect Research*, In Proceedings of the 15th ACM Conference on Knowledge and Information Management (CKIM), 2006.
- [24] Zhou, D., Manavoglu, E. Li, J., Giles, L., Zha, H., *Probabilistic Models for Discovering E-Communities*, In Proceedings of the 15th ACM International World Wide Web Conference (WWW), 2006.
- [25] http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_uts/stop_words