LDA-Based Document Models for Ad-hoc Retrieval

Xing Wei and W. Bruce Croft
Computer Science Department
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003
{xwei,croft}@cs.umass.edu

ABSTRACT

Search algorithms incorporating some form of topic model have a long history in information retrieval. For example, cluster-based retrieval has been studied since the 60s and has recently produced good results in the language model framework. An approach to building topic models based on a formal generative model of documents, Latent Dirichlet Allocation (LDA), is heavily cited in the machine learning literature, but its feasibility and effectiveness in information retrieval is mostly unknown. In this paper, we study how to efficiently use LDA to improve ad-hoc retrieval. We propose an LDA-based document model within the language modeling framework, and evaluate it on several TREC collections. Gibbs sampling is employed to conduct approximate inference in LDA and the computational complexity is analyzed. We show that improvements over retrieval using cluster-based models can be obtained with reasonable efficiency.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Retrieval models*.

General Terms

Theory, Experimentation

Keywords

Information Retrieval, Language Model, Latent Dirichlet Allocation (LDA), Topic Model, Document Model.

1. INTRODUCTION

Representing the content of text documents is a critical part of any approach to information retrieval (IR). Typically, documents are represented as a "bag of words", meaning that the words are assumed to occur independently. To capture important relationships between words, researchers have proposed approaches that group words into "topics". Techniques such as word clustering and document clustering have been used for many years to enhance document representations. Word or term clustering, for example, was studied in the 60s (Sparck Jones, 1971). The well-known Latent Semantic Indexing (LSI) technique was introduced in 1990 (Deerwester et al, 1990). More recently, Hoffman (1999) described the probabilistic Latent Semantic Indexing (pLSI) technique. This approach uses a latent variable model that represents documents as mixtures of topics. Although Hoffman showed that pLSI outperformed LSI in a vector space model framework, the data sets used were small and not representative of modern IR environments. Specifically, the collections in these experiments only contained a few thousand document abstracts.

Using topic models for document representation has also recently been an area of considerable interest in machine learning. Latent Dirichlet Allocation or LDA (Blei et al, 2003), has quickly become one of the most popular probabilistic text modeling techniques in machine learning and has inspired a series of research papers (e.g., Girolami and Kaban, 2005; Teh et al, 2004). LDA has been shown to be effective in some text-related tasks such as document classification, but the feasibility and effectiveness of using LDA in IR tasks remains mostly unknown. Possessing fully generative semantics, LDA potentially overcomes the drawbacks of previous topic models such as pLSI (Hoffman, 1999). Language modeling (Ponte and Croft, 1998; Berger and Lafferty, 1999), which is one of the most popular statistically principled approaches to IR, is also a generative model, motivating us to examine LDA-based document representations in the language modeling framework.

The LDA approach will be compared with an approach that builds topic models using document clusters, known in the machine learning literature as the mixture of unigrams model (McCallum, 1999). Liu and Croft (2004) showed that document clustering can improve retrieval effectiveness in the language modeling framework. Retrieval based on cluster models (referred to here as cluster-based retrieval) performed consistently well across several TREC collections, and significant improvements over document-based retrieval models were reported. In the language modeling framework, the cluster-based topic models were used to smooth the probabilities in the document model (Liu and Croft, 2004). As a much simpler topic model, the mixture of unigrams model generates a whole document from one topic under the assumption that each document is related to exactly one topic. This assumption may, however, be too simple to effectively model a large collection of documents. In contrast, LDA models a document as a mixture of multiple topics.

Given the potential advantages of LDA as a generative model of documents, and the encouraging results with topic models in previous work, we carried out a detailed evaluation of

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.
SIGIR'06, August 6-11, 2006, Seattle, WA, USA.
Copyright 2006 ACM 1-58113-000-0/00/0004…$5.00.

the effectiveness of LDA-based retrieval in large collections. Azzopardi et al. (2004) also discussed the applications of LDA models and reported inconclusive results on several small collections. In this paper, we propose an LDA-based document model for IR, evaluate it on TREC collections, and discuss efficiency issues. In Section 2, we discuss related work in topic model based retrieval, including pLSI and the cluster model. We present the new retrieval model based on LDA in Section 3, and compare the complexity of LDA with the clustering algorithms used in Liu and Croft (2004) in Section 4. We then describe the data sets and experimental methods in Section 5. Retrieval performance is discussed in Section 6. Finally, Section 7 concludes and discusses possible directions for future work.

2. RELATED WORK

2.1 Probabilistic Latent Semantic Indexing (pLSI)

The probabilistic Latent Semantic Indexing model, which was introduced by Hoffman (2003) quickly gained acceptance in a number of text modeling applications. pLSI, also called an aspect model, is a latent variable model for general co-occurrence data which associates an unobserved class (topic) variable with each observation (i.e., with each occurrence of a word). The roots of pLSI go back to Latent Semantic Indexing/Analysis (Deerwester et al, 1990). pLSI was designed as a discrete counterpart of LSI to provide a better fit to text data. It can also be regarded as an attempt to relax the assumption made in the mixture of unigrams model that each document is generated from a single topic. pLSI models each document as a mixture of topics. The following process generates documents in the pLSI model:

1) Pick a topic mixture distribution $P(.|d)$ for each document d,

2) Pick a latent topic z with probability $P(z|d)$ for each word token,

3) Generate the word token w with probability $P(w|z)$.

The probability of generating a document d, as a bag of words $w_1 ... w_{N_d}$ (N_d is the number of words of document d), is:

$$P(w_1 ... w_{N_d}) = \prod_{i=1}^{N_d} \sum_{z=1}^{K} P(w_i \mid z) P(z \mid d) \qquad (1)$$

Hoffman (1999) applied pLSI to retrieval tasks in the Vector Space Model framework, albeit on small collections. He exploited pLSI both as a unigram model to smoothen the empirical word distributions and as a latent space model to provide a low-dimensional document/query representation. Significantly better retrieval performance over the standard term matching method based on the raw term frequencies and Latent Semantic Indexing (LSI) was reported on all four collections, which contained 1033, 1400, 3204, and 1460 document abstracts. The smoothing parameter was optimized by hand for each collection.

Although large improvements were reported, the collection sizes and the document lengths in the collections are far from representative of realistic IR environments, making the effectiveness of the mixture-of-topics model on IR tasks still unclear. In addition, the baseline retrieval model was far from state-of-the-art. The pLSI model itself has a problem in that its generative semantics are not well-defined (Blei et al, 2003); thus there is no natural way to predict a previously unseen document, and the number of parameters of pLSI grows linearly with the number of training documents, which makes the model susceptible to overfitting.

2.2 Cluster-based Retrieval

The cluster model, also known as the mixture of unigrams model, has been well examined in IR research. In the cluster model, it is assumed that all documents fall into a finite set of K clusters (topics). Documents in each cluster discuss a particular topic z, and each topic z is associated with a multinomial distribution $P(w|z)$ over the vocabulary. The process of generating a document d ($w_1 ... w_{N_d}$) in the cluster model is as follows:

1) Pick a topic z from a multinomial distribution with parameter θ_z

2) For $i = 1 ... N_d$, pick word w_i from topic z with probability $P(w_i |z)$.

The overall likelihood of observing the document d from the cluster model is:

$$P(w_1 ... w_{N_d}) = \sum_{z=1}^{K} P(z) \prod_{i=1}^{N_d} P(w_i \mid z) \qquad (2)$$

One of the parameter estimation methods for the mixture of unigrams model is to cluster documents in the collection into K groups and then use a maximum likelihood estimate a topic model $P(w|z)$ for each cluster. Liu and Croft (2004) adopted this method with a K-means clustering algorithm. They incorporated the cluster information into language models as smoothing:

$$P(w \mid D) = \frac{N_d}{N_d + \mu} P_{ML}(w \mid D) + (1 - \frac{N_d}{N_d + \mu}) P(w \mid cluster) \qquad (3)$$

With the new document model they conducted experiments on several TREC collections, finding that cluster-based retrieval performs consistently across collections. Significant improvements over document-based retrieval are obtained.

The cluster model possesses fully generative semantics, but the assumption that each string (document) is generated from a single topic is limiting and may become problematic for long documents and large collections.

3. LDA-BASED DOCUMENT MODEL

3.1 Latent Dirichlet Allocation

As we described in Section 2.1, the pLSI model has a problem with inappropriate generative semantics. Blei et al. (2003) introduced a new, semantically consistent topic model, Latent Dirichlet Allocation (LDA), which immediately attracted a considerable interest from the statistical machine learning and natural language processing communities. The basic generative process of LDA closely resembles pLSI. In pLSI, the topic mixture is conditioned on each document. In LDA, the topic mixture is drawn from a conjugate Dirichlet prior that remains the same for all documents. The process of generating a corpus is as follows (we consider the smoothed LDA here):

1) Pick a multinomial distribution ϕ_z for each topic z from a Dirichlet distribution with parameter β;

2) For each document d, pick a multinomial distribution θ_d from a Dirichlet distribution with parameter α,

3) For each word token w in document d, pick a topic $z \in \{1 \ldots K\}$ from the multinomial distribution θ_d,

4) Pick word w from the multinomial distribution ϕ_z.

Thus, the likelihood of generating a corpus is:

$$P(Doc_1,...,Doc_N \mid \alpha,\beta) \\ = \iint \prod_{z=1}^{K} P(\phi_z \mid \beta) \prod_{d=1}^{N} P(\theta_d \mid \alpha) (\prod_{i=1}^{N_d} \sum_{z_i=1}^{K} P(z_i \mid \theta) P(w_i \mid z, \phi)) d\theta d\phi \quad (4)$$

The LDA model is represented as a probabilistic graphical model in Figure 1.

Compared to the pLSI model, LDA possesses fully consistent generative semantics by treating the topic mixture distribution as a k-parameter hidden random variable rather than a large set of individual parameters which are explicitly linked to the training set; thus LDA overcomes the overfitting problem and the problem of generating new documents in pLSI.

Compared to the cluster model, LDA allows a document to contain a mixture of topics, relaxing the assumption made in the cluster model that each document is generated from only one topic. This assumption may be too limited to effectively model a large collection of documents; in contrast, the LDA model allows a document to exhibit multiple topics to different degrees, thus being more flexible.

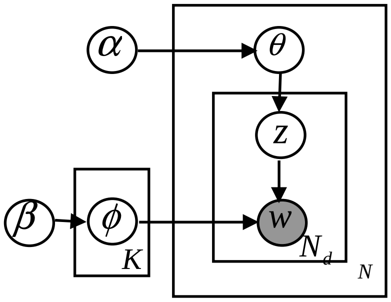

Figure 1. Graphical model representation of LDA. *K* is the number of topics; *N* is the number of documents; N_d is the number of word tokens in document *d*.

3.2 LDA-based Retrieval

The basic approach for using language models for IR is the query likelihood model where each document is scored by the likelihood of its model generating a query Q,

$$P(Q \mid D) = \prod_{q \in Q} P(q \mid D) \quad (5)$$

where D is a document model, Q is the query and q is a query term in Q. $P(Q|D)$ is the likelihood of the document model generating the query terms under the 'bag-of-words' assumption that terms are independent given the documents. $P(q_i \mid D)$ is specified by the document model with Dirichlet smoothing (Zhai and Lafferty, 2001),

$$P(w \mid D) = \frac{N_d}{N_d + \mu} P_{ML}(w \mid D) + (1 - \frac{N_d}{N_d + \mu}) P_{ML}(w \mid coll) \quad (6)$$

where $P'(w|D)$ is the maximum likelihood estimate of word w in the document D, and $P'(w|coll)$ is the maximum likelihood estimate of word w in the entire collection. μ is the Dirichlet prior, and in our reported experiments we used a fixed value with μ=1000 since the best results are consistently obtained with this setting.

Document modeling (estimating $P(w|D)$) is crucial to retrieval. Compared to the standard query likelihood model, LDA offers a new and interesting framework to model documents. However, as in other topic models, a topic in the LDA model represents a combination of words; and it may not be as precise a representation as words in non-topic models like the unigram model. Therefore LDA itself (commonly used with a relatively limited number of topics) may be too coarse to be used as the only representation for IR. Indeed, our preliminary experiments show that directly employing the LDA model hurts retrieval performance. So, we instead combine the original document model (Eqn. 6) with the LDA model and construct a new LDA-based document model. Motivated by the significant improvements obtained by Liu and Croft (2004), we formulate our model through a linear combination obtained in one of the following ways: (a) linearly combining the original document model and LDA, which is illustrated in (7), (b) additively combining the LDA model with the maximum likelihood estimate of word w in the document D, and (c) combining the LDA model with the Dirichlet smoothing part, i.e. the maximum likelihood estimate of word w in the entire collection. Option (c) is similar to the combination used in Liu and Croft (2004). All methods have empirically shown similar performance with appropriate parameters, and we will only report results of Option (a) which performs slightly better in our experiments (parameter setting in our paper is for (a); it may be necessary to adjust λ and μ in (b) and (c)).

$$P(w \mid D) = \lambda (\frac{N_d}{N_d + \mu} P_{ML}(w \mid D) + (1 - \frac{N_d}{N_d + \mu}) P_{ML}(w \mid coll)) \\ + (1 - \lambda) P_{lda}(w \mid D) \quad (7)$$

The LDA model has a new representation for a document based on topics. After we get the posterior estimates of θ and ϕ, we can calculate the probability of a word in a document as following,

$$P_{lda}(w \mid d, \hat{\theta}, \hat{\phi}) = \sum_{z=1}^{K} P(w \mid z, \hat{\phi}) P(z \mid \hat{\theta}, d) \quad (8)$$

where $\hat{\theta}$ and $\hat{\phi}$ are the posterior estimates of θ and ϕ respectively.

The LDA model is very complex and cannot be solved by exact inference. There are a few approximate inference techniques available in the literature: variational methods (Blei et al, 2003), expectation propagation (Griffiths and Steyvers, 2004) and Gibbs sampling (Geman and Geman, 1984; Griffiths and Steyvers, 2004). We use Gibbs sampling and the approximation of $\widehat{\theta}$ and $\widehat{\phi}$ can be obtained directly. From a Gibbs sample, we use $(n_{-i,j}^{(w_i)} + \beta_{w_i}) / \sum_{v=1}^{V} (n_{-i,j}^{(v)} + \beta_v)$ to approximate $\widehat{\phi}$ and $(n_{-i,j}^{(d_i)} + \alpha_{z_i}) / \sum_{t=1}^{T} (n_{-i,t}^{(d_i)} + \alpha_t)$ to approximate $\widehat{\theta}$ after a certain number of iterations (burn-in period) being accomplished, where $n_{-i,j}^{(w_i)}$ is the number of instances of word w_i assigned to topic *z=j*, not including the current token, α and β are hyper-parameters that determine how heavily this empirical distribution is smoothed, and can be chosen to give the desired resolution in the resulting distribution, $n_{-i,j}^{(d_i)}$ is the number of words in document d_i (the document that token *i* belongs to) assigned to topic *z=j*, not including the current token. Thus $\sum_{v=1}^{V} n_{-i,j}^{(v)}$ is the total number of words assigned to topic *z=j*; and $\sum_{t=1}^{T} n_{-i,t}^{(d_i)}$ is the total number of words in document d, not including the current one (Griffiths and Steyvers, 2004). Thus (7) will be

$$P(w \mid D) = \lambda\left(\frac{N_d}{N_d + \mu} P'(w \mid D) + \left(1 - \frac{N_d}{N_d + \mu}\right) P'(w \mid coll)\right) + (1-\lambda)\left(\sum_{t=1}^{K} \frac{n_{-i,j}^{(w_i)} + \beta_{w_i}}{\sum_{v=1}^{V} (n_{-i,j}^{(v)} + \beta_v)} \times \frac{n_{-i,j}^{(d_i)} + \alpha_{z_i}}{\sum_{t=1}^{T} (n_{-i,t}^{(d_i)} + \alpha_t)}\right) \quad (9)$$

Although Eqn.(9) involves the approximated posterior distribution using one Gibbs sample, we can use the samples from different Markov chains with different initializations. Our experiment shows that using multi-Markov chains is useful. So the actual value of $P_{lda}(w \mid D)$ we used is an average of the ones from several Markov Chains.

3.3 Complexity

Complexity is often a big concern for topic models. Even the simple cluster model suffers from potentially high computational costs. Liu and Croft (2004) used a three-pass K-means algorithm primarily motivated by its efficiency. They showed that the running time for each pass/iteration grows linearly with the number of documents (*N*) and the number of classes (*K*), i.e., *O(KN)*. Roughly speaking, the complexity of each iteration of the Gibbs sampling for LDA is also linear with the number of topics/clusters and the number of documents, which is also *O(KN)*. Due to the large sizes of document collections, we give a more detailed analysis.

The time-consuming part of the Gibbs sampling in the LDA model is linear with *I*, *K* and $N * \overline{N}_t$, where *I* is the number of iterations, *K* is the number of topics, *N* is the number of documents and $\overline{N}_t$ is the average number of tokens in one document. In K-means clustering algorithm, the computation is linear with *I*, *N*, and $K * \overline{N}_w$, where *I* is the number of passes/iterations, and $\overline{N}_w$ is the average number of unique terms in one cluster. (We use the average numbers, $\overline{N}_t$ and $\overline{N}_w$, instead of the corresponding sums to make the following comparison easier.)

To compare the running time of these two algorithms we compare realistic values of these items.

(1) *K*: The selected number of topics (*K*) in the LDA model is generally less than the selected number of topics/clusters in the cluster model because in the LDA model topics can be mixed to represent one document, but in the cluster model one document can based on only one topic.

(2) *I*: The number of iterations (I) will probably have a larger value in the LDA algorithm. In Liu and Croft (2004), the number of iterations for K-means is 3. Such a small *I* does not work well for Gibbs sampling in the LDA model. The selection of *I* is very important to make sure that the Markov chains reach equilibrium. In Section 4.3.1, we show that N_i = 30 ~ 50 is reasonable in our experiments.

(3) $\overline{N}_t$ vs. $\overline{N}_w$: It is hard to make an assertion about the relationship of these two items, especially since $\overline{N}_w$ is highly related to the selection of *K*. While in our experiments and settings, the number of unique terms in a cluster is often larger than $\overline{N}_t$ since one cluster often contains quite many documents.

The above comparison shows that the efficiency of the two algorithms is similar. In experiments, we also find that the difference in running times between LDA and K-means is trivial. Based on our experience based on using several IR collections, these two algorithms are comparable in computational costs and there is no clear evidence showing that one algorithm is obviously more efficient.

4. EXPERIMENTS AND RESULTS

4.1 Data

We conducted experiments on five data sets taken from TREC: the Associated Press Newswire (AP) 1988-90 with queries 51-150, Wall Street Journal (WSJ) 1987-92 with queries 51-100 and 151-200, Financial Times (FT) 1991-94 with queries 301-400, San Jose Mercury News (SJMN) 1991 with queries 51-150, and LA Times (LA) with queries 301-400. Queries are taken from the "title" field of TREC topics. Relevance judgments are taken from the judged pool of top retrieved documents by various participating retrieval systems from previous TREC conferences. Queries that have no relevant documents in the judged pool for a specific collection have been removed from the query set for that collection. Statistics of the collections and query sets are given in Table 1.

These five collections, including the query sets and relevance judgments, were the same as used by Liu and Croft (2004) in order to compare LDA-based retrieval with cluster-based retrieval. The only difference between the two experimental settings is that we left out the Federal Register (FR) collection for two reasons: (1) The query set of this collection contains only 21 valid queries[1], (the query sets of other collections contain around 100 (>=94) valid queries); (2) In these 21 valid queries there are six that have only one relevant document in the collection and thus may cause biased results.

[1] "Valid queries" means queries that have relevant docs.

Table 1. Statistics of data sets

Collection	Contents	# of dos	Size	Queries	# of Queries with Relevant Docs
AP	Associated Press newswire 1988-90	242,918	0.73Gb	TREC topics 51-150 (title only)	99
FT	Financial Times 1991-94	210,158	0.56Gb	TREC topics 301-400 (title only)	95
SJMN	San Jose Mercury News 1991	90,257	0,29Gb	TREC topics 51-150 (title only)	94
LA	LA Times	131,896	0.48Gb	TREC topics 301-400 (title only)	98
WSJ	Wall Street Journal 1987-92	173,252	0.51Gb	TREC topics 51-100 & 151-200 (title only)	100

4.2 Parameters

There are several parameters that need to be determined in our experiments. For the retrieval experiments, the proportion of the LDA part in the linear combination must be specified (λ in (6)). For the LDA estimation, the number of topics must be specified; the number of iterations and the number of Markov chains also need to be carefully tuned due to its influence on performance and running time. We use the AP collection as our training collection to estimate the parameters. The WSJ, FT, SJMN, and LA collections are used for testing whether the parameters optimized on AP can be used consistently on other collections. At the current stage of our work, the parameters are selected through exhaustive search or manually hill-climbing search. All parameter values are tuned based on average precision since retrieval is our final task. The parameter selection process, including the training set selection, also follows Liu and Croft (2004) to make the results comparable. Mean average precision is used as the basis of evaluation throughout this study.

We use symmetric Dirichlet priors in the LDA estimation with $\alpha = 50/K$ and β =0.01, which are common settings in the literature. Our experience shows that retrieval results are not very sensitive to the values of these parameters.

4.2.1 LDA Estimation

Document models consisting of mixtures of topics, like pLSI and LDA, have previously been tested mostly on small collections due to their relatively long running time. From Section 3.3 it is shown that the iteration number in LDA estimation plays an important role in its complexity. Generally, more iterations means that the Markov chain reaches equilibrium with higher probability, and after a certain number of iterations (burn-in period) the invariant distribution of the Markov chain is equivalent to the true distribution. So it would be ideal if we could take samples right after the Markov chain reach equilibrium. However, in practice, convergence detection of Markov chains is still an open research question. That is, no realistic method can be applied on the large IR collections to determine the convergence of the chain. Researchers in the area of topic modeling tend to use a large number of iterations to guarantee convergence. However, in IR tasks it is almost impossible to run a very large number of iterations due to the size of the data set. Besides, a finely tuned topic model does not naturally mean good retrieval performance. Instead, a less accurate distribution of topics may be good enough for IR purposes. Furthermore, we have λ and μ in our model to adjust the influence of the LDA model. For example, if the LDA estimation is coarse, we may reduce the smoothing weight and let the LDA estimation share a part of smoothing.

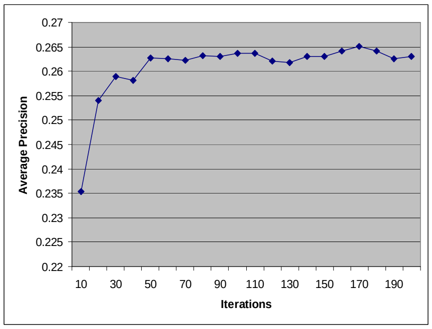

Figure 2. Retrieval results (in average precision) on AP with different number of iterations. *K*=400; λ =0.7; 1 Markov chain.

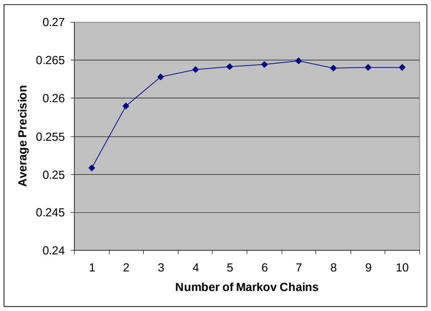

Figure 3. Retrieval results (in average precision) on AP with different number of Markov chains. *K*=400; λ =0.7; 30 iterations.

In order to get a good iteration number that is effective for IR applications, we use the AP collection for training and maximizing the average precision score as the optimization criterion since it is our final evaluation metric. We try different iteration numbers, and also do experiments with different numbers of Markov chains, each of which is initialized with a different random number, to see how many chains are needed for our purposes. The results are presented in Figure 2 and Figure 3, respectively. After 50 iterations and with more than 3 Markov chains, performance is quite stable, so we use these values in the final retrieval experiments. The running time of each iteration with large topic numbers can be expensive; 30 iterations and 2 chains are a good trade off between accuracy and running time, and these values are used in the parameter-selecting experiments, especially when selecting a suitable number of topics.

Selecting the right number of topics is also an important problem in topic modeling. Nonparametric models like the Chinese Restaurant Process (Blei et al, 2004; Teh et al, 2004) are not practical to use for large data sets to automatically decide the number of topics. A range of 50 to 300 topics is typically used in the topic modeling literature. 50 topics are often used for small collections and 300 for relatively large collections, which are still much smaller than the IR collections we use. It is well known that larger data sets may need more topics in general, and it is confirmed here by our experiments with different values of *K* (100, 200, …) on the AP collection. *K*=800 gives the best average precision, as shown in Table 2. This number is much less than the corresponding optimal *K* value (2000) in the cluster model (Liu and Croft, 2004). As we explained in Section 3.3, in the cluster model, one document can be based on one topic, and in the LDA model, the mixture of topics for each document is more powerful and expressive; thus a smaller number of topics is used. Empirically, even with more parsimonious parameter settings like *K*=400, 30 iterations, 2 Markov chains, statistically significant improvements can also be achieved on most of the collections.

Table 2. Retrieval results (in average precision) on AP with different number of topics (*K*).

K	**100**	**200**	**300**	**400**	**500**
Average precision	0.2431	0.2520	0.2579	0.2590	0.2557
600	**700**	**800**	**900**	**1000**	**1500**
0.2578	0.2609	0.2621	0.2613	0.2585	0.2579

4.2.2 Parameters in Retrieval Model

In order to select a suitable value of λ, we use a similar procedure as above on the AP collection and find 0.7 to be the best value in our search. From the experiments on the testing collections, we also find that λ=0.7 is the best value or almost the best value for other collections.

We set the Dirichlet prior μ=1000 since the best results are consistently obtained with this setting. The value of μ needs to be adjusted when the other combination methods discussed in Section 3.2 are applied.

4.3 Experimental Results

In all experiments, both the queries and documents are stemmed, and stopwords are removed.

4.3.1 Retrieval Experiments

The retrieval results on the AP collection are presented in Table 3, with comparisons to the result of query likelihood retrieval (QL) and cluster-based retrieval (CBDM). Statistically significant improvements of LDA-based retrieval (LBDM) over both QL and CBDM are observed at many recall levels, with 21.64% and 13.97% improvement in average precision respectively.

Table 3. Comparison of query likelihood retrieval (QL), cluster-based retrieval (CBDM) and retrieval with the LDA-based document models (LBDM). The evaluation measure is average precision. AP data set. Stars indicate statistically significant differences in performance with a 95% confidence according to the Wilcoxon test.

	QL	**CBDM**	**LBDM**	**%chg over QL**	**%chg over CBDM**
Rel.	21819	21819	21819		
Rel. Retr.	10130	10751	12064	+10.09*	+12.21*
0.00	0.6422	0.6485	0.6795	+5.8*	+4.8*
0.10	0.4339	0.4517	0.4844	+11.6*	+7.2*
0.20	0.3477	0.3713	0.4131	+18.8*	+11.2*
0.30	0.2977	0.317	0.3661	+23.0*	+15.5*
0.40	0.2454	0.2668	0.311	+26.8*	+16.6*
0.50	0.2081	0.2274	0.2666	+28.1*	+17.2*
0.60	0.1696	0.1794	0.2245	+32.4*	+25.1*
0.70	0.1298	0.1444	0.1665	+28.3*	+15.3*
0.80	0.0865	0.1002	0.118	+36.5*	+17.8*
0.90	0.0480	0.0571	0.0694	+44.7	+21.6
1.00	0.0220	0.0201	0.0187	-15.1	-6.8
Avg	0.2179	0.2326	0.2651	+21.64*	+13.97*

With the parameter setting λ =0.7, 50 iterations and 3 Markov chains, we run experiments on other collections and present results in Table 4. We compare the results with CBDM, and the results of the query likelihood model are also listed as a reference. On all five collections, LDA-based retrieval achieves improvements over both of query likelihood retrieval and cluster-model based retrieval, and four of the improvements are significant (over CBDM). Considering that CBDM has already obtained significant improvements over the query likelihood model (and Okapi-style weighting, see Liu and Croft) on all of these collections, and is therefore a high baseline, the significant performance improvements from LBDM are very encouraging.

Unlike the basic document representation, the LDA-based document model is not limited to only the literal words in a document, but instead describes a document with many other related highly probable words from the topics of this document. For example, for the query “buyout leverage”, the document “AP900403-0219”, which talks about “Farley Unit Defaults On Pepperell Buyout Loan”, is a relevant document. However, this document focuses on the “buyout” part, and does not contain the exact query term “leverage”, which makes this document rank very low. Using the LDA-based representation, this document is closely related to two topics that have strong connections with the term “leverage”: one is the *economic* topic that is strongly associated with this document because the document contains many representative terms of this topic, such as “million”, “company”, and “bankruptcy”; the other is the *money market* topic which is closely connected to “bond”, also a very frequent word in this document. In this way, the document is ranked higher with the LDA-based document model. Having multiple topics represent a document tends to give a clearer association between words than the single topic model used in cluster-based retrieval.

Table 4. Comparison of cluster-based retrieval (CBDM) and retrieval with the LDA-based document models (LBDM). The evaluation measure is average precision. %chg denotes the percentage change in performance (measured in average precision) of LBDM over QL and CBDM. Stars indicate statistically significant differences in performance between LBDM and QL/CBDM with a 95% confidence according to the Wilcoxon test.

Collection	QL	CBDM	LBDM	%chg over QL	%chg over CBDM
AP	0.2179	0.2326	0.2651	+21.64*	+13.97*
FT	0.2589	0.2713	0.2807	+7.54*	+3.46*
SJMN	0.2032	0.2171	0.2307	+13.57*	+6.26*
LA	0.2468	0.2590	0.2666	+8.02[2]	+2.93
WSJ	0.2958	0.2984	0.3253	+9.97*	+9.01*

4.3.2 Comparison and Combination with Relevance Models

In Table 5 we compare the retrieval results of the LBDM with the relevance model (RM), which incorporates pseudo-feedback information and is known for excellent performance (Lavrenko and Croft, 2001). On some collections, the results of the two models are quite close. RM uses pseudo-feedback information and thus needs *online* processing, i.e., it effectively does an extra search for each query, which makes it less efficient in reacting to users' inputs. As an *offline*-processing model that does not do any extra processing on queries, the LDA-based retrieval model performance is quite impressive. In another words, we estimate the LDA model offline only once, and then LBDM can process real-time queries much more efficiency than RM with similar performance.

Table 5. Comparison of the relevance models (RM) and the LDA-based document models (LBDM). The evaluation measure is average precision. %diff indicates the percentage change of LBDM over RM.

Collection	QL	LBDM	RM	%diff
AP	0.2179	0.2651	0.2745	-3.42
FT	0.2589	0.2807	0.2835	-0.99
SJMN	0.2032	0.2307	0.2633	-12.38
LA	0.2468	0.2666	0.2614	+0.20
WSJ	0.2958	0.3253	0.3422	-4.94

The improvement on the AP collection in Table 4 is relatively larger than on the other collections. Although we tune parameters on the AP collection, parameter adjustment for the other collections does not improve the performance much. Compared to the relevance model results in Table 5, we conjecture that it is due to the characteristics of the documents and queries that the improvement on the AP collection is larger than on the other collections.

We can also combine the relevance model and LBDM to do retrieval. In this case, the retrieval results using LBDM are used as the pseudo-feedback for the relevance model. Results are shown in Table 6, and results of the query likelihood model are also listed as a reference. Moderate improvements are obtained, which are better than the very small improvements reported in Liu and Croft (2004) for the combination of RM and CBDM.

Table 6. Comparison of the relevance model (RM) and the combination of RM and the LDA-based document model (RM+LBDM). The evaluation measure is average precision. %chg denotes the percentage change in performance (measured in average precision) of RM+LBDM over RM. Stars indicate statistically significant differences in performance between RM+LBDM and RM with a 95% confidence according to the Wilcoxon test.

Collection	QL[3]	RM	RM+LBDM	%chg
AP	0.2161	0.2758	0.2869	+4.00
FT	0.2558	0.2889	0.2907	+0.62
SJMN	0.1985	0.2547	0.2603	+2.22
LA	0.2290	0.2509	0.2715	+8.21
WSJ	0.2908	0.3405	0.3606	+5.91*

5. CONCLUSIONS AND FUTURE WORK

We have proposed LDA-based document models for ad-hoc retrieval, and evaluated the method using several TREC collections. Based on the experimental results, we can make the following conclusions. Firstly, experiments performed in the language modeling framework, including combination with the relevance model, have demonstrated that the LDA-based document model consistently outperforms the cluster-based approach, and the performance of LBDM is close to the Relevance Model, which incorporates pseudo-feedback information. Secondly, we have shown that the estimation of the LDA model on IR tasks is feasible with suitable parameters based on the analysis of the algorithm complexity and empirical parameter selections. More importantly, unlike the Relevance Model, LDA estimation is done offline and only needs to be done once. Therefore LDA-based retrieval can potentially be used in applications where pseudo-relevance feedback would not be

[2] This improvement is significant according to t-test, and almost significant (with a 93% confidence) according to the Wilcoxon test.

[3] The QL&RM baseline in Table 6 is slightly different with Table 5 because in the experiments of Table 5, in order to compare with the results in Liu and Croft (2004), we directly load their index into our system and then run the experiments on their index to get nearly identical results.

possible. In summary, LDA-based retrieval is a promising method for IR, although more work needs to be done with even larger collections, such as the Web data from the TREC Terabyte track.

For future work, we have begun to investigate whether other topic models (e.g. Griffiths et al, 2005; Wei and McCallum, 2006) that have recently been developed can further improve retrieval performance. An approximation that can improve LDA estimation will also be helpful. In addition, we plan to re-examine some traditional topic modeling methods (i.e. term clustering) as to their efficiency and effectiveness in retrieval tasks.

6. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by Advanced Research and Development Activity and NSF grant #CCF-0205575, and in part by NSF grant #IIS-0527159. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

7. REFERENCES

Azzopardi, L., Girolami, M and van Rijsbergen, C.J. Topic Based Language Models for ad hoc Information Retrieval. In *Proceedings of the International Joint Conference on Neural Networks*, Budapest,Hungary, 2004.

Berger, A. and Lafferty, J. Information Retrieval as Statistical Translation. In *Proceedings of ACM SIGIR 99*, 222-229, 1999.

Blei, D. M., Ng, A. Y., and Jordan, M. J. Latent Dirichlet allocation. In *Journal of Machine Learning Research*, 3, 993-1022, 2003.

Blei, D., Griffiths, T., Jordan, M., Tenenbaum, J. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, MIT Press, 2004.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391-407, 1990.

Geman, S., and Geman, D. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-741, 1984.

Girolami, M. and Kaban, A. Sequential activity profiling: latent Dirichlet allocation of Markov chains. *Data Mining and Knowledge Discovery*, 10, 175–196, 2005.

Girolami, M. and Kaban, A. On an equivalence between PLSI and LDA. In *Proceedings of SIGIR '03*, 433-434, 2003.

Griffiths, T. L., and Steyvers, M. Finding scientific topics. In *Proceeding of the National Academy of Sciences.* 5228-5235, 2004.

Griffiths, T. L., Steyvers, M., Blei, D. and Tenenbaum, J. Integrating topics and syntax. In *Advances in Neural Information Processing Systems* 17, 2005

Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of SIGIR '99*, Berkeley, CA, USA, 1999.

Lavrenko, V. and Croft, W. B. Relevance-based language models. In *Research and Development in Information Retri*eval, 120-127, 2001.

Li, W. and McCallum, A. DAG-Structured Mixture Models of Topic Correlations. To appear in *Proceedings of the 23rd International Conference on Machine Learning (ICML-06),* Pittsburgh, Pennsylvania, USA, 2006.

Liu, X. and Croft, W. B. Cluster-based retrieval using language models. In *Proceedings of SIGIR '04*, 186-193, 2004..

McCallum, A. Multi-label text classification with a mixture model trained by EM. In A*AAI workshop on Text Learning*. 1999.

Ponte, J. and Croft, W.B. A language modeling approach to information retrieval. In *Proceedings of ACM SIGIR 1998*, 275-281, 1998.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence.* Banff, Alberta, Canada, 2004.

Sparck Jones, K. *Automatic keyword classification for information retrieval*. Butterworths, London, 1971.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. *Hierarchical Dirichlet processes.* Technical Report, Department of Statistics, UC Berkeley, 2004.

Zhai, C. and Lafferty, J. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of ACM SIGIR 2001*, 334-342, 2001.