A Comparative Study of Utilizing Topic Models for Information Retrieval

Xing Yi and James Allan

Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, Amherst, MA 01003-4610, USA

Abstract. We explore the utility of different types of topic models for retrieval purposes. Based on prior work, we describe several ways that topic models can be integrated into the retrieval process. We evaluate the effectiveness of different types of topic models within those retrieval approaches. We show that: (1) topic models are effective for document smoothing; (2) more rigorous topic models such as Latent Dirichlet Allocation provide gains over cluster-based models; (3) more elaborate topic models that capture topic dependencies provide no additional gains; (4) smoothing documents by using their similar documents is as effective as smoothing them by using topic models; (5) doing query expansion should utilize topics discovered in the top feedback documents instead of coarse-grained topics from the whole corpus; (6) generally, incorporating topics in the feedback documents for building relevance models can benefit the performance more for queries that have more relevant documents.

Keywords: Topic Model, Retrieval, Evaluation.

1 Introduction

Topic models are a very popular approach for representing the content of documents. A document is assumed to draw its vocabulary from one or more topics. Topics are represented as probability distributions over the vocabulary, where differing topics give different words high probabilities. We can automatically infer a set of topics either by simple clustering[1] or methods popularized by the machine learning community [2,3,4]. These topics can be used to describe the contents of a collection: the high probability topics and words within the topics can be viewed as a loose description of the collection, with better topic models providing better descriptions. A natural question is whether these topics are useful to help retrieve documents on the same topic as a query – intuitively relevant documents have topic distributions that are likely to have generated the set of words associated with the query[2,5]. In fact, early research on topic models suggested that they might be used for information retrieval (IR)[5,6], but it was not until recently that they were successfully applied to large-scale and realistic collections [7]. Others have claimed that topic models can improve IR by matching queries to documents at a semantic level [8].

Our goal in this study is to explore the utility of different types of topic models for retrieval purposes. If more sophisticated topic models better reflect

M. Boughanem et al. (Eds.): ECIR 2009, LNCS 5478, pp. 29–41, 2009.

the content of the collection, then they should be more useful when retrieving documents. Furthermore, by representing a document as mixture of topics, complicated topics models may help to discover some fine-grained topics related to a query in the relevant documents. To investigate these issues, we describe several ways in which topic models have been incorporated into retrieval.

We cannot possibly study all topic modeling approaches, so we select a few that are representative: the well-known Mixture of Unigrams (MU) model [1]; Latent Dirichlet Allocation (LDA) [2], a more complicated and computationally expensive topic model; and Pachinko Allocation Model (PAM) [3], a recently proposed new topic model which not only models the relations between words and identifies topics but also models the organization and co-occurrences of the topics themselves. We also select an unusual "topic modeling" approach, Relevance Modeling (RM) [9], that treats each document as the representative of its own topic. Finally we include the model-based feedback (MFB) approach [10] which assumes feedback documents related to a query are generated through mixing a query-related topic and the background topic.

We start this study in § 2 by describing in more detail the topic models that we have selected. Then we describe different ways for incorporating topic models into IR, including both document and query expansion in § 3. We next evaluate the approaches on different types of topic models using TREC retrieval tasks in § 4. Despite using a wide range of topic models and mechanisms for incorporating them into retrieval, we find that the RM approach consistently outperforms more elaborate topic modeling methods. This result is weaker for topics with large numbers of relevant documents. We conclude that topic models are only likely to be useful when query topics are very broad, even when mixtures of topic models are used to represent finer-grained topics.

2 Topic Models

In this section, we briefly review several approaches to creating topic models given a collection of documents. Space considerations prevent us from providing more than a sketch in most cases. We stress that we are not inventing new topic modeling techniques in this paper. We start with a set of definitions that will be used through the remainder of this paper. Each word w takes values in the vocabulary $\mathcal{V}$. Each document D is a sequence of N_D words denoted by $D = (w_1, w_2, ..., w_{N_D})$. Each corpus $\mathcal{C}$ is a collection of M documents denoted by $\mathcal{C} = (D_1, D_2, ..., D_M)$. Each topic t_i in a topic model TM is parameterized as a multinomial distribution over words in the vocabulary – $\{P(w|t_i), w \in \mathcal{V}\}$.

2.1 Statistical Topic Models

Statistical or probabilistic topic models are generative processes that specify procedures by which documents are created [8]. There are a range of topic models,

but a broad outline of document generation is: pick some topics and then, for each word in the document, pick a topic from that set and select a word from the topic. We review these models by discussing their differences in generating documents and calculating the document generating probabilities $p(D)$, which are important for understanding topic model based retrieval methods.

The unigram model assumes that the words of every document D are drawn independently from a single multinomial distribution; thus, there is only a single topic t in the whole corpus $\mathcal{C}$. In contrast, the Mixture of Unigrams (MU) model [4] assumes that there are multiple topics in the corpus and each document is assigned to one of those topics. Given a multinomial distribution $\theta = (\theta_1, ..., \theta_k)$ over k topics, each document D is generated by first sampling a topic t_i from θ then sampling N_D words independently from the multinomial distribution $P(w|t_i)$; therefore, we have:

$$p(D|\theta) = \sum_{t_i} p(t_i|\theta) \prod_{n=1}^{N_D} p(w_n|t_i) = \sum_{t_i} \theta_{t_i} \prod_{n=1}^{N_D} p(w_n|t_i). \quad (1)$$

Latent Dirichlet Allocation (LDA) [2] is a widely-used topic model which also assumes that there are multiple topics in the corpus but that a document can have multiple topics. LDA has a more complicated probabilistic procedure of generating a document. Essentially, given a distribution over topics, the words in a document are generated by first selecting a topic from that distribution and then selecting a word from that topic.

Although LDA captures correlations among words, it does not explicitly model the correlations among topics. In contrast, a recently-proposed topic model – Pachinko Allocation Model (PAM) [3] explicitly captures the topic correlations by sampling over *super-topics* – mixture of topics. In this paper, we consider the four-level PAM, which consists of root-topic node, super-topic nodes, sub-topic nodes and word nodes. LDA can be viewed as a three-level PAM consisting of only the root-topic node, topic nodes and word nodes. The document generation process is similar in spirit but incorporates the topic hierarchy rather than an unordered collection of topics.

To train MU, we utilized an efficient document clustering approach [1], which first clusters documents by using any clustering algorithm like K-means, then estimates a multinomial distribution for each cluster. To train LDA and PAM, because exact inference is intractable, we utilized the Gibbs Sampling approach [3,7] for approximate inference. The training complexity of different inference methods have been analyzed elsewhere [3,4,7]. Note that training sophisticated topic models like LDA and PAM is much more computationally expensive than training MU. For example, in our experiments, using the WSJ corpus (173,252 documents) to train topic models and running on a computer having Intel(R) Xeon(TM) 3.2GHz CPU, 4GB memory and Linux OS: an 800-topics MU took about 8 hours to finish 25 iterations (converged); an 800-topics LDA took 7 days and 14 hours to finish 1000 iterations (converged); an 800 sub-topics and 100 super-topics PAM took 18 days and 6 hours to finish 100 iterations (not converged).

2.2 Topic Models from IR

The Relevance Modeling (RM) [9] approach assumes the following process to generate a string $(w, q_1...q_k)$ given a query $q = \{q_1...q_k\}$: first sample a document D_i in the whole corpus $\mathcal{C}$, then sample $k+1$ times from its distribution $p(w|D_i)$. We can follow the same process to generate a document D that has N_D words and have:

$$p(D) = \sum_{D_i \in \mathcal{C}} p(D_i) \prod_{n=1}^{N_D} p(w_n|D_i). \tag{2}$$

Comparing this equation with Equation (1), it can be seen that the generative process in RM can be viewed as an unusual MU topic modeling approach that treats each document D_i as the representative of its own topic t_i.

The model-based feedback (MFB) approach [10] assumes feedback documents related to a query q are generated through a two-component mixture model, of which one component is the background topic $\theta_{\mathcal{C}}$ and the other component is a query dependent topic θ_q. To generate each feedback document, for each word, MFB first picks either θ_q or $\theta_{\mathcal{C}}$ to generate this word, then samples the word from the selected topic. Because MFB allows each document to have two mixed topics, it is different from MU while similar to LDA. Formally, the document generating probability is:

$$p(D) = \prod_{n=1}^{N_D} (\gamma p(w_n|\theta_q) + (1-\gamma)p(w_n|\theta_{\mathcal{C}})), D \in \mathcal{F}, \tag{3}$$

where γ is the probability of sampling θ_q to generate the given word w_n and fixed to be a constant, $\mathcal{F}$ is the feedback document set. $\theta_{\mathcal{C}}$ is typically fixed to be the unigram model trained with the whole collection $\mathcal{C}$, representing background information or even non-relevant topics, θ_q is estimated through EM algorithm[10].

3 Document Retrieval

There are two obvious approaches to including topic models in IR. In the first, a document is represented by itself and the topics to which it belongs, which means that $P(w|D)$ is calculated by somehow incorporating probabilities in topics. A second approach is to calculate a query related topic by using topic models and use it for query expansion. In each case, there are different options for merging the documents or queries with the topics.

3.1 Topic Model Based Document Models

Document model smoothing techniques [11] use the probability of a word in the whole corpus $p(w|\mathcal{C})$ to smooth the maximum likelihood (ML) estimate of observing a word in a document $p_{ML}(w|D)$, thus obtaining a better document language model $p(w|D)$. When using Dirichlet smoothing, we have:

$$p(w|D) = \frac{N_D}{N_D + \mu} p_{ML}(w|D) + \frac{\mu}{N_D + \mu} p(w|\mathcal{C}), \tag{4}$$

which can be used in the typical query likelihood approach for retrieval. We call this baseline **QL**.

Given a topic model TM, any document D and word $w, w \in \mathcal{V}$, we first calculate a topic model based document model $p_{TM}(w|D)$ by:

$$p_{TM}(w|D) = \sum_{t_i \in \mathcal{T}} p_{TM}(w|t_i) p_{TM}(t_i|D), \tag{5}$$

where $p_{TM}(w|t_i)$ is the multinomial distribution in topic t_i, $p_{TM}(t_i|D)$ is the probability of observing topic t_i in D, and $\mathcal{T}$ represents the topic set utilized to calculate this document model. $\mathcal{T}$ can either contain all the topics in model TM or just one topic t_{best} that a document D belongs to with the highest probability: $t_{best} = \arg\max_{t_i} p_{TM}(t_i|D)$. After that, $p_{TM}(w|D)$ is combined with $p(w|\mathcal{C})$ for smoothing in order to calculate a better document model $p'(w|D)$ for retrieval:

$$p'(w|D) = \alpha p_{ML}(w|D) + \beta p(w|\mathcal{C}) + (1 - \alpha - \beta) p_{TM}(w|D); \tag{6}$$

When making different choices of topic models, document smoothing techniques and $\mathcal{T}$, Equation (6) can result in different retrieval methods, including some recently proposed topic model based IR models:

1. Let $\mathcal{T}$ contain only t_{best} in MU. First use Jelinek-Mercer (JM) smoothing to smooth $p_{MU}(w|D)$ with $p(w|\mathcal{C})$, then use Dirichlet smoothing to smooth the $p_{ML}(w|D)$, which is the cluster-based document modeling (**CBDM**) retrieval method [12]:

$$p'(w|D) = \tfrac{N_D}{N_D+\mu} p_{ML}(w|D) + \tfrac{\mu}{N_D+\mu}[\lambda p_{MU}(w|D) + (1-\lambda) p(w|\mathcal{C})] \tag{7}$$

2. Let $\mathcal{T}$ contain all topics in LDA. Use Dirichlet smoothing to smooth $p_{ML}(w|D)$ with $p(w|\mathcal{C})$, then further smooth the result with $p_{LDA}(w|D)$, which is the LDA-based document modeling (**LBDM**) retrieval method [7]:

$$p'(w|D) = (1-\lambda)(\tfrac{N_D}{N_D+\mu} p_{ML}(w|D) + \tfrac{\mu}{N_D+\mu} p(w|\mathcal{C})) + \lambda p_{LDA}(w|D). \tag{8}$$

3. Let $\mathcal{T}$ contain all topics in RM. Use Dirichlet smoothing to smooth $p_{ML}(w|D)$ with $p(w|\mathcal{C})$, then further smooth the result with $p_{RM}(w|D)$:

$$p'(w|D) = (1-\lambda)(\tfrac{N_D}{N_D+\mu} p_{ML}(w|D) + \tfrac{\mu}{N_D+\mu} p(w|\mathcal{C})) + \lambda p_{RM}(w|D). \tag{9}$$

From the view of RM, $p_{RM}(w|D)$ is the relevance model [9] of the document D by using D as the query; thus, this relevance model based document expansion (**RMDE**) is in fact doing document expansion, which is similar to another document expansion based retrieval method (**DELM**) [13].

In this framework, we can also design new topic based retrieval methods. For example, when smoothing the document with its highest ranked topic as in the CBDM method and using the topic from LDA, PAM and RM, we have retrieval methods – **BT-LBDM**, **BT-PBDM** and **RMDE-1**, respectively; when smoothing the document with a weighted combination of all topics that it contains as the LBDM method and using the topics from MU, PAM and RM, we have **MBDM**, **PBDM** and **RMDE**, respectively.

3.2 Topic Model for Query Expansion

There are multiple ways of using topic models to calculate a query-specific topic θ_q – a multinomial distribution $p(w|q)$ for a given query $q = \{q_1, ..., q_k\}$, for query expansion. The MFB approach[10] employs two-component mixture models to directly estimate θ_q from feedback documents retrieved by the query, while the RM [9] approach calculates θ_q by using each posterior $p(D_i|q)$ of document D_i generating query q and the document topic $p(w|D_i)$:

$$p(w|q) = \sum_{D_i \in \mathcal{C}} p(w|D_i) \times p(D_i|q); \tag{10}$$

To investigate whether topics t_is discovered in the whole corpus $\mathcal{C}$ can be used for query expansion, we follow the RM approach by replacing the document D_i with these topics, and calculate a topic model based relevance model:

$$p_{TM}(w|q) = \sum_{t_i} p_{TM}(w|t_i) \times p_{TM}(t_i|q). \tag{11}$$

Intuitively, this approach ranks each topic t_i by its probability of generating the query q, then uses the words in high ranked topics to calculate a query-specific topic for query expansion. Then by using different models, we can have a family of topic model based query expansion retrieval methods: when using topics from MU, LDA, PAM, we have **CBQE**, **LBQE**, **PBQE**, respectively.

To investigate whether topics discovered by typical topic models in the feedback documents can be directly used for query expansion, we train topic models with the top-k documents retrieved by a query, calculate a set of topics and plug them into above equation (11) to calculate a query-specific topic for query expansion. We call methods by this approach **Q-CBQE** and **Q-LBQE** when using MU and LDA, respectively. Note that from this aspect, RM can be viewed as one special case of Q-CBQE where the number of topics is equal to the number of feedback documents.

Furthermore, to investigate whether the multiple topic representation of document by sophisticated topic models like LDA can be directly combined into the RM approach to calculate a better relevance model, we extend equation (10) as:

$$p(w|q) = \sum_{D_i \in \mathcal{C}} (\gamma p_{RM}(w|D_i) + (1-\gamma) p_{TM}(w|D_i, q)) \times p(D_i|q), \tag{12}$$

where γ is a constant to control relative portions of probability from the original RM $p_{RM}(w|D_i)$ and from a topic model $p_{TM}(w|D_i, q)$ used to calculate a better relevance model. We further assume given a topic t_m, each word w is generated independently with other words, thus we have $p_{TM}(w|t_m, D_i, q) = p_{TM}(w|t_m)$ and $p_{TM}(q|t_m, D_i) = p_{TM}(q|t_m)$, then:

$$p_{TM}(w|D_i, q) = \sum_{t_m} p(w|t_m) \times p(t_m|D_i, q), \quad p(t_m|D_i, q) = p(t_m, D_i, q)/p(D_i, q) = \\ p(t_m)p(D_i|t_m)p(q|t_m)/p(D_i, q) = p(t_m|D_i)p(q|t_m)/p(q|D_i) \propto p(t_m|D_i)p(q|t_m). \tag{13}$$

Intuitively, equation (13) can be explained as when using the topic models to calculate a relevance model for a given query q, each topic portion $p(t_m|D_i)$ in

the document should be weighted by the probability of this topic generating the query $p(q|t_m)$. We call this complicated approach of combining topic models for calculating a relevance model as TM-RM, e.g. when using LDA, it is called **LDA-RM**.

4 Experiments

Five TREC corpora and the corresponding TREC *ad hoc* retrieval tasks in Table 1 are used for comparing different approaches of using topic model for IR. The queries are taken from the "title" field of TREC topics. These data were previously used for studying performance of CBDM and LBDM [7,12]. We also use their same training/testing split experimental settings for comparison: using the AP corpus as the training collection to tune parameters and the other four corpora (FT, SJMN, LA, WSJ) for testing.

When applying topic models on the whole corpus and using topic models for document smoothing, the number of topics for MU and LDA is tuned to be 2000 and 800 respectively, which are the settings that perform the best by CBDM and LBDM. For methods using PAM, to reduce the number of tuning parameters and also to compare with LDA, we use 800 sub-topics and 100 super-topics, and tune other parameters. We train MU and LDA to convergence, but only train PAM with 100 burn-in iterations because of limited computational resources (recall it took more than 18 days for the 100 iterations). The Dirichlet smoothing parameter μ is tuned to be 1000 for all methods, the JM smoothing parameter λ is tuned to be different values for different methods.

For each method in the query expansion approach, after calculating a query-specific topic θ_q or a topic model based relevance model $p_{TM}(w|q)$, we follow other researchers [7,10] to smooth it with the original query model $p(w|q)$:

$$p'_{TM}(w|q) = \lambda p_{TM}(w|q) + (1-\lambda)p(w|q)., \tag{14}$$

then use the $p'_{TM}(w|q)$ is for retrieval by using cross entropy [14] as the ranking measurement. λ is tuned for each method. For the methods using topics from the whole corpus for query expansion like CBQE, LBQE and PBQE, we tune the number of top ranked topics for calculating relevance models; for the methods using topics from the top-k feedback documents, we tune the number k ; we also tune the number of topics used in the topic models like Q-LBQE. For the MFB, we set $\gamma = 0.5$ as suggested [10]; for the LDA-RM, we further tune γ.

Table 1. Statistics of TREC corpora and topics. Net topics indicates the number that had relevant documents. All topics are title only.

Collection	Contents	# of Docs	Size	Topics	Net topics
AP	Associated Press 1998-90	242,918	0.73Gb	51-150	99
FT	Financial Times 1991-4	210,158	0.56Gb	301-400	95
SJMN	San Jose Mercury News 1991	90,257	0.29Gb	51-150	94
LA	LA Times	131,896	0.48Gb	301-400	98
WSJ	Wall Street Journal 1987-92	173,252	0.51Gb	51-100,151-200	100

Table 2. Retrieval Performance with TREC topics 301-400 (title-only) on one testing corpus (FT) by using different topic models for query expansion and for document smoothing. There are overall 3233 relevant documents. Bold font highlights the best result in each column. Parameters tuned on the training corpus for using typical topic models on the top feedback documents are not well generalized to this FT testing corpus: Q-CBQE, Q-LBQE, MFB perform worse than the QL baseline.

	Rel.	Interpolated Recall - Precision					Precision:			MAP
	Retr.	0.00	0.10	0.20	0.40	0.60	P@5	P@10	P@100	
QL	1879	0.6142	0.4615	0.3987	0.2989	0.2136	0.3747	0.3242	0.1117	0.2614
CBDM	2092	0.6057	0.4766	0.4106	0.3042	0.2234	0.3768	0.3221	0.1144	0.2738
BT-LBDM	2074	0.6082	0.4821	0.4068	0.3062	0.2153	0.3705	0.3211	0.1142	0.2681
BT-PBDM	2034	0.6147	0.4747	0.4127	0.2952	0.2187	0.3789	0.3126	0.1144	0.2675
RMDE-1	1946	0.6067	0.4832	**0.4329**	0.3400	0.2344	0.3726	0.3284	0.1178	0.2836
MBDM	2099	0.5983	0.4810	0.4076	0.3058	0.2169	0.3705	0.3200	0.1137	0.2718
LBDM	2216	0.6338	0.4899	0.4072	0.3213	0.2329	0.3705	0.3147	0.1227	0.2787
PBDM	2226	**0.6341**	**0.4993**	0.4201	0.3207	0.2382	**0.3958**	0.3200	0.1229	0.2823
RMDE	2134	0.6320	0.4914	0.4294	0.3212	0.2334	0.3726	0.3221	0.1207	0.2811
CBQE	2016	0.6067	0.4681	0.4005	0.3210	0.2063	0.3537	0.3053	0.1166	0.2634
LBQE	2007	0.6198	0.4770	0.4034	0.3092	0.2114	0.3663	0.3168	0.1179	0.2663
PBQE	1981	0.6203	0.4648	0.3917	0.2983	0.2090	0.3642	0.3074	0.1159	0.2607
Q-CBQE	2151	0.5638	0.4517	0.3719	0.3015	0.2186	0.3516	0.3032	0.1254	0.2544
Q-LBQE	2028	0.5856	0.4671	0.3709	0.2928	0.2085	0.3411	0.2863	0.1193	0.2541
MFB	2283	0.5351	0.4303	0.3658	0.2793	0.2112	0.3347	0.2979	0.1254	0.2469
LDA-RM	2266	0.6113	0.4874	0.4295	0.3432	0.2576	0.3663	0.3263	0.1276	0.2947
RM	**2313**	0.6103	0.4844	0.4326	**0.3592**	**0.2626**	0.3768	**0.3389**	**0.1295**	**0.3006**

4.1 Results and Analysis

Table 2 shows the best retrieval results on one of the four testing corpora (FT). Our results of CBDM and LBDM are only slightly different from earlier results [7,12] due to small differences in the implementations. Table 3 further shows the pair-wise significance test results of the MAP differences between some well-performed methods and other methods on the FT corpus. MAP results on the other testing corpora (WSJ, SJMN and LA) and the tuning corpus (AP) are shown in Table 4.

We have the following observations: (1) Using topic models for document smoothing can improve IR performance of the typical smoothing technique; complicated topic models like LDA and PAM have some benefits: LBDM and PBDM achieve higher MAPs than CBDM on every corpus. (2) The document expansion approach RMDE, which borrows idea from RM to do document smoothing and does not actually identify topics in the collection, usually performs better than CBDM, and sometimes similar to LBDM. (3) LBDM performs usually better than PBDM although PAM is more powerful for topic representation; thus, for retrieval, more complicated topic models may not bring further improvement. (4) Topic models trained with the whole corpus are too coarse-grained to be useful for query expansion. (5) Topic models trained with the query dependent feedback documents can perform extremely well on the training corpus; however, they are sensitive to the tuned parameters and not always well generalized

Table 3. Significance tests of the difference between MAPs of some methods on the testing corpus (FT). For each column, stars and triangles in each cell indicate the column method has statistically significant higher MAP than the row method according to the Wilcoxon test and one-sided t-test ($p < 0.05$) respectively.

	LBDM	PBDM	LDA-RM	RMDE-1	RMDE	RM
QL	⋆	⋆△	⋆△	⋆△	⋆△	⋆△
CBDM			⋆		⋆	⋆△
BT-LBDM	⋆△	⋆△	⋆△	⋆△	⋆△	⋆△
BT-PBDM	⋆	⋆△	⋆△	⋆△	⋆△	⋆△
RMDE-1			⋆			⋆
MBDM	⋆	⋆	⋆△		⋆	⋆△
LBDM						⋆△
PBDM						
RMDE			⋆			⋆△
CBQE			⋆△	⋆		⋆△
LBQE			⋆△	△	⋆	⋆△
PBQE	⋆	⋆△	⋆△	△	⋆△	⋆△
Q-CBQE			⋆△			⋆△
Q-LBQE			⋆△			⋆△
MFB	⋆△	⋆△	⋆△	△	△	⋆△
LDA-RM						

Table 4. MAPs of different methods on the tuning corpus AP and testing corpora WSJ, SJMN and LA. Bold font shows the 1st and 2nd best results for each corpus.

	AP	WSJ	SJMN	LA		AP	WSJ	SJMN	LA
QL	0.2142	0.2646	0.1612	0.2275	CBQE	0.2368	0.2628	0.1710	0.2206
CBDM	0.2332	0.2738	0.1802	0.2298	LBQE	0.2286	0.2701	0.1656	0.2194
BT-LBDM	0.2329	0.2662	0.1771	0.2330	PBQE	0.2243	0.2666	0.1666	0.2151
BT-PBDM	0.2260	0.2738	0.1715	0.2207	Q-CBQE	**0.2856**	0.3035	0.1948	0.2400
RMDE-1	0.2235	0.2794	0.1774	0.2457	Q-LBQE	0.2633	0.2979	0.1880	0.2333
MBDM	0.2360	0.2771	0.1842	0.2316	MFB	**0.2833**	0.3116	0.1965	0.2302
LBDM	0.2608	0.2819	0.1989	0.2499	LDA-RM	0.2830	**0.3229**	**0.2094**	**0.2565**
PBDM	0.2466	0.2815	0.1908	0.2382	RM	0.2775	**0.3264**	**0.2116**	**0.2605**
RMDE	0.2399	0.2841	0.1784	0.2436					

to other testing corpora. (6) Although RM does not perform the best on the training corpus, it performs consistently well on different testing corpora. (7) LDA-RM, which aims at combining the advantages from LDA's multiple topic representation for documents and RM's viewing each document as its own topic, does well on both training and testing corpora.

To further improve RM's performance, researchers have proposed to use topic model based document smoothing retrieval methods like CBDM and LBDM instead of the simple QL to get better feedback documents[7,12], so that better relevance models can be built for a second round retrieval. Their approaches achieved very small improvement. To investigate the impact of training topic

Table 5. MAPs of different approaches of combining RM and topic modeling. Stars and triangles indicate significant improvement on RM according to the Wilcoxon test and t-test ($p < 0.05$), respectively.

	AP	FT	WSJ	SJMN	LA
RM	0.2775	0.3006	0.3264	0.2116	0.2605
LBDM-LDA-RM	0.2982⋆△	0.3048	0.3372	0.2211⋆	0.2651
LBDM-RM	0.2953⋆△	0.3088 ⋆	0.3377 △	0.2229⋆	0.2705

models with better feedback documents, we use the LBDM instead of QL in the first round retrieval, then employ LDA-RM to build a relevance model for a second round retrieval; we also employ the typical RM in the second round retrieval for comparison. The former method is denoted as **LBDM-LDA-RM** and the latter one as **LBDM-RM**. The MAP results are shown in Table 5. We can see that although on the training corpus both combination methods perform significantly better than RM, the significantly better results only exist in some testing corpora.

We also observe that although the LDA-RM approach of building relevance models helps on the training corpus (LDA-RM better than RM, LBDM-LDA-RM better than LBDM-RM), using the typical RM approach in the second round retrieval still wins on all testing corpora. To investigate in which specific case incorporating topic models trained on the feedback documents can help, we first calculate the per query average precision (AP) difference $\triangle AP_q$ between the LBDM-RM and LBDM-LDA-RM, i.e. $\triangle AP_q = AP_{LBDM-LDA-RM,q} - AP_{LBDM-RM,q}$. Then we consider the relation between the $\triangle AP_q$ of a query and its number of relevant documents: intuitively, a query with few relevant documents usually makes it hard for topic models to discover query-related topics because they have limited number of relevant training documents. For this analysis, we divide each query set in Table 1 into four equal-sized subsets according to each query's number of relevant documents, then calculate the maxima, minima, averages and medians of the $\triangle AP_q$s of the queries in each quartile of the query set in each retrieval task. The results are shown in Table 6. We point out that different retrieval tasks have very different characteristics: the median number of relevant documents per query in the AP, FT, WSJ, SJMN or LA retrieval task is 119, 15, 72, 32 or 14 respectively.

We have the following observations from Table 6: (1) Both approaches of building relevance models have their advantages and disadvantages and the $\triangle AP_q$ of each query varies a lot especially when it does not have many relevant documents in the corpus – the $1st$ quartile has the largest min-max spread of $\triangle AP_q$. (2) Fewer number of relevant documents hurt the performance of the LDA-RM approach more than the RM approach – the averages and medians of the $\triangle AP_q$s in the $1st$ and $2nd$ quartiles of each retrieval task are lower than in the $3rd$ and $4th$ quartiles. (3) More relevant documents can reduce the performance gap between the two approaches and benefit the topic modeling approach more, although the improvement is small – the averages and medians of the $\triangle AP_q$s are usually increasing when more and more relevant documents are available. We

Table 6. Some statistics of the $\triangle AP_q$s of the queries in each quartile of the query set in each retrieval task. The $1st$ or $4th$ quartile contains 25% queries that have the fewest or largest number of relevant documents in each retrieval task respectively.

AP	1st	2nd	3rd	4th quartile	FT	1st	2nd	3rd	4th quartile
max	0.0044	0.0236	0.0220	0.3754	max	0.0533	0.0255	0.0705	0.1460
min	-0.0534	-0.0186	-0.0128	-0.0067	min	-0.1280	-0.0580	-0.0435	-0.0277
avg.	-0.0071	-0.0005	0.0015	0.0177	avg.	-0.0128	-0.0074	-0.0004	0.0065
median	-0.0015	-0.0016	0.0003	0.0019	median	-0.0005	-0.0017	-0.0003	0.0004
WSJ					SJMN				
max	0.0583	0.0522	0.0491	0.0209	max	0.1257	0.0274	0.0376	0.0131
min	-0.0455	-0.0545	-0.0181	-0.0179	min	-0.0553	-0.0333	-0.0215	-0.0275
avg.	-0.0013	-0.0027	0.0019	0.0003	avg.	-0.0026	-0.0036	-0.0018	0.0007
median	0.0000	-0.0028	-0.0009	0.0007	median	0.0000	-0.0011	-0.0022	0.0018

conclude that although our complicated topic modeling approach of building relevance models does have its own advantage, the typical RM approach performs more robustly when queries have limited number of relevant documents, which makes the average performance of RM better in these TREC *ad hoc* retrieval tasks where queries generally do not have many relevant documents. Our finding also supports Lavrenko's earlier argument [15] that RM is more appealing on handling rare events than aspect-based topic models like LDA.

5 Conclusions

In this paper, we compare the utility of different types of topic models for IR. A general topic model based retrieval framework has been presented, which covers two different approaches: document model smoothing and query expansion. Previous work of using topic models for IR has been discussed in this framework, and new topic models like PAM can be easily used for retrieval in this framework. For the query expansion approach, we have discussed several ways of calculating a query-specific topic either from feedback documents or from the whole corpus, and then introduced the TM-RM (in experiment, LDA-RM) approach, which combines advantages from topic models' capability of representing documents as mixture of topics and RM's viewing a document as its own specific topic to discover a better query-specific topic.

We evaluate different topic model based retrieval methods by using the TREC *ad hoc* retrieval tasks. Experimental results show that training topic models with the whole corpus and using them for document smoothing can improve IR performance over a simple document smoothing approach. More powerful and complicated model like PAM does not necessary provide further IR benefits than LDA. In addition, a document expansion approach (RMDE), which does not actually identify topics in the collection, performs well and sometimes similar to using LDA for smoothing.

Topics discovered in the whole corpus are too coarse-grained to be useful for query expansion. Topics discovered in the query related feedback documents

can help retrieval, although performances of many methods using these query related topics for retrieval are sensitive to parameters and not always perform well for different retrieval tasks. RM performs consistently well in both training and testing corpora and out-performs most topic modeling approaches. The complicated TM-RM approach also performs consistently well and successfully improves some queries' results, compared with the RM approach; however, its average performance is still a little worse than the RM approach. To investigate why this happens, we compare the per query performance difference between two approaches. We find that the RM approach performs more robustly when queries have limited number of relevant documents while the TM-RM approach works better in the case that a query has more relevant documents—i.e., that a query's relevant documents match the broadness of a topic.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, and in part by UpToDate. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

1. Xu, J., Croft, W.B.: Cluster-based language models for distributed retrieval. In: Proceedings of ACM SIGIR, pp. 254–261 (1999)
2. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. Journal of machine Learning Research 3, 993–1022 (2003)
3. Li, W., McCallum, A.: Pachinko Allocation: DAG-structured mixture models of topic correlations. In: Proceedings of ICML, Pittsburgh, PA, pp. 577–584 (2006)
4. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39(2/3), 103–134 (2000)
5. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of ACM SIGIR, Berkeley,CA,USA, pp. 50–57 (1999)
6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
7. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. In: Proceedings of ACM SIGIR, Seattle, Washington, pp. 178–185 (2006)
8. Steyvers, M., Griffiths, T.: Probabilistic topic models. Handbook of Latent Semantic Analysis (2007)
9. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: Proceedings of ACM SIGIR, pp. 120–127 (2001)
10. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of CIKM, pp. 403–410 (2001)
11. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad-hoc Information Retrieval. In: Proceedings of ACM SIGIR, pp. 334–342 (2001)

12. Liu, X., Croft, W.B.: Cluster-based retrieval using language models. In: Proceedings of ACM SIGIR, Sheffield, UK, pp. 186–193 (2004)
13. Tao, T., Wang, X., Mei, Q., Zhai, C.: Language model information retrieval with document expansion. In: Proceedings of HLT/NAACL, pp. 407–414 (2006)
14. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for Information Retrieval. In: Proceedings of ACM SIGIR, pp. 111–119 (2001)
15. Lavrenko, V.: A generative theory of relevance. Ph.D. Dissertation, 55–56 (2004)