A Biterm Topic Model for Short Texts

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, Xueqi Cheng
Institute of Computing Technology, CAS
Beijing, China 100190
yanxiaohui@software.ict.ac.cn, {guojiafeng, lanyanyan, cxq}@ict.ac.cn

ABSTRACT

Uncovering the topics within short texts, such as tweets and instant messages, has become an important task for many content analysis applications. However, directly applying conventional topic models (e.g. LDA and PLSA) on such short texts may not work well. The fundamental reason lies in that conventional topic models implicitly capture the document-level word co-occurrence patterns to reveal topics, and thus suffer from the severe data sparsity in short documents. In this paper, we propose a novel way for modeling topics in short texts, referred as *biterm topic model (BTM)*. Specifically, in BTM we learn the topics by directly modeling the generation of word co-occurrence patterns (i.e. biterms) in the whole corpus. The major advantages of BTM are that 1) BTM explicitly models the word co-occurrence patterns to enhance the topic learning; and 2) BTM uses the aggregated patterns in the whole corpus for learning topics to solve the problem of sparse word co-occurrence patterns at document-level. We carry out extensive experiments on real-world short text collections. The results demonstrate that our approach can discover more prominent and coherent topics, and significantly outperform baseline methods on several evaluation metrics. Furthermore, we find that BTM can outperform LDA even on normal texts, showing the potential generality and wider usage of the new topic model.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval; I.5.3 [**Pattern Recognition**]: Clustering

Keywords

Short Text, Topic Model, Biterm, Content Analysis, document clustering

1. INTRODUCTION

Short texts are prevalent on the Web, no matter in traditional Web sites, e.g. Web page titles, text advertisements and image captions, or in emerging social media, e.g. tweets, status messages, and questions in Q&A websites. Uncovering the topics of such short texts is crucial for a wide range of content analysis tasks, such as content characterizing [26, 35, 14], user interest profiling [32], emerging topic detecting [20] and so on. However, unlike the traditional normal documents (e.g. news articles and academic papers), the lack of rich context in short texts makes the topic modeling a challenging problem.

Conventional topic models, like PLSA [16] and LDA [3], are widely used for uncovering the hidden topics from text corpus. In general, documents are modeled as mixtures of topics, where a topic is a probability distribution over words. Statistical techniques are then utilized to learn the topic components and mixture coefficients of each document. In essence, the conventional topic models reveal the latent topics within the text corpus by implicitly capturing the document-level word co-occurrence patterns [5, 30]. Therefore, directly applying these models on short texts will suffer from the severe data sparsity problem (i.e. the sparse word co-occurrence patterns in each short document) [17]. More specifically, 1) the occurrences of words in short document play less discriminative role compared to lengthy documents where the model has enough word counts to know how words are related [17] ; 2) The limited contexts make it more difficult for topic models to identify the senses of ambiguous words in short documents.

One simple but popular way to alleviate the sparsity problem is to aggregate short texts into lengthy pseudo-documents before training a standard topic model. For example, Weng et al. [32] aggregated the tweets published by individual user into one document before training LDA. Besides the user-based aggregation, Hong et al. [17] also aggregated the tweets containing the same word, and shown that topic models trained on these aggregated messages work better than the regular LDA. However, such heuristic data aggregation methods are highly data-dependent. For example, the user information is not always available in some datasets, like the collection of Web page titles or advertisements. Even if the user information is available, e.g. in tweets data, most users only have few tweets which makes the aggregation less effective.

Another way to deal with the problem is to make stronger assumptions on the data. A typical way is to assume that a short document only covers a single topic. For example, Zhao et al. [35] modeled each tweet in the way of mixture of unigrams [23]. Similar approach can be found in [12], where words in each sentence are assumed to be drawn from the same topic. Compared to LDA and PLSA, the simplified data generation process may help alleviate the sparsity problem in short texts. However, it loses the flexibility to capture different topic ingredients in one document, and suf-

WWW, '13 Rio Ode Karo, Brazil
ACM 978-1-4503-2035-1/13/05.

fers from overfitting issues due to the peaked posteriors of topics P(z|d) [3].

Unlike these approaches, in this paper, we propose a novel topic model for short texts to tackle the sparsity problem. The main idea comes from the answers of the following two questions. 1) Since topics are basically groups of correlated words and the correlation is revealed by word co-occurrence patterns in documents, why not explicitly model the word co-occurrence for topic learning? 2) Since topic models on short texts suffer from the problem of severe sparse patterns in short documents, why not use the rich global word co-occurrence patterns for better revealing topics?

Specifically, we propose a generative *biterm topic model (BTM)*, which learns topics over short texts by directly modeling the generation of biterms in the whole corpus. Here, a *biterm* is an unordered word-pair co-occurred in a short context. The data generation process under BTM is that the corpus consist of a mixture of topics, and each biterm is drawn from a specific topic. Compared with conventional topic models, the major differences and advantages of BTM lie in that 1) BTM explicitly models the word co-occurrence patterns (i.e. biterms), rather than documents, to enhance the topic learning; and 2) BTM uses the aggregated patterns in the whole corpus for learning topics to solve the problem of sparse patterns at document-level. By learning BTM, we can obtain the topic components and a global topic distribution of the corpus, except the topic distribution of each individual document as it does not model the document generation process. However, we show that the topic distribution of each document can be naturally derived based on the learned model.

We conduct extensive experiments on two real-world short text collections, i.e. the datasets from Twitter and a Q&A website. Experimental results show that BTM can discover more prominent and coherent topics than the baseline methods. Quantitative evaluations confirm the superiority of BTM on several evaluation metrics. Additionally, we also test our approach on a normal text collection, i.e. 20Newsgroup. It is surprising for us to find that BTM can outperform LDA even on normal texts, showing the potential generality and wider usage of the new topic model.

The rest of the paper is organized as follows: in Section 2, we give a brief review of related works. Section 3 introduces our model for short text topic modeling, and discuss its implementation in Section 4. Experimental results are presented in Section 5. Finally, conclusions are made in the last section.

2. RELATED WORKS

In this section, we briefly summarize the related work from the following two perspectives: topic models on normal texts, and that on short ones.

2.1 Topic Models on Normal texts

Topic models have been proposed to uncover the latent semantic structure from text corpus. The effort of mining semantic structure in a text collection can be dated from latent semantic analysis (LSA) [9], which utilizes the singular value decomposition of the document-term matrix to reveal the major associative words patterns. Probabilistic latent semantic analysis (PLSA) [16] improves LSA with a sounder probabilistic model based on a mixture decomposition derived from a latent class model. In PLSA, a document is presented as a mixture of topics, while a topic is a probability distribution over words. Extending PLSA, Latent Dirichlet Allocation (LDA) [3] adds Dirichlet priors on topic distributions, resulting in a more complete generative model. Due to its nice generalization ability and extensibility, LDA achieves huge success in text mining domain.

In the last decade, topic models have been extensively studied. Many more complicated variants and extensions of LDA and PLSA have been proposed, such as the author-topic model [27], Bayesian nonparametric topic model [29], and supervised topic model [2]. Among them two works close to us are the recently proposed regularized topic model [22] and the generalized Pólya model [21], which also employ word co-occurrence statistics to enhance topic learning. However, both of them utilize word co-occurrences as structure priors for topic-word distribution, rather than directly modeling their generation process. Above all, almost all the models mentioned above deal with normal text without considering the specificity of short texts.

2.2 Topic Models on Short Texts

Early studies mainly focused on exploiting external knowledge to enrich the representation of short texts. For example, Sahami et al.[28] suggested a search-snippet-based similarity measure for short texts. Phan et al.[24] learned hidden topics from large external resources to enrich the representation of short texts. Jin et al.[19] learned topics on short texts via transfer learning from auxiliary long text data. These ways may be helpful in some specific domains, but not general since favorable external dataset might not be always available. Additionally, these approaches and ours are complementary rather than competitive.

With the emergence of social media in recent years, topic models have been utilized for social media content analysis in various tasks, such as content characterizing [26, 35], event tracking [20], content recommendation [25, 8], and influential users prediction [32]. However, due to the lack of specific topic models for short texts, some researchers directly applied conventional (or slightly modified) topic models for analysis [26, 31]. Some others tried to aggregate short texts into lengthy pseudo-documents based on some additional information, and then train conventional topic models [32, 35]. Hong et al. [17] made a comprehensive empirical study of topic modeling in Twitter, and suggested that new topic models for short texts are in demand.

In our previous works, we developed methods based on non-negative matrix factorization for short text clustering [34] and topic learning [33] by exploiting global word co-occurrence information. This work extends them by proposing a more principle approach to model topics over short texts. To the best of our knowledge, the proposed topic model is the first one focusing on general-domain short texts, which does not exploit any external knowledge.

3. OUR APPROACH

Conventional topic models learn topics based on document-level word co-occurrence patterns, whose effectiveness will be highly influenced in short text scenario where the word co-occurrence patterns become very sparse in each document. To tackle this problem, here we propose a novel biterm topic model, which learns topics over short texts by directly modeling the generation of all the biterms (i.e. word co-occurrence patterns) in the whole corpus.

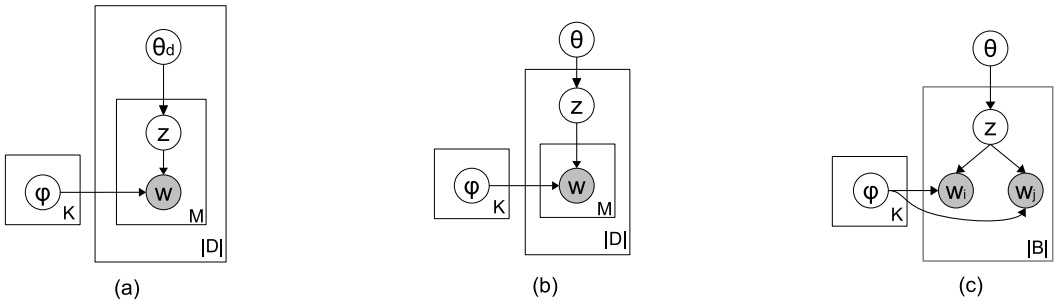

Figure 1: Graphical representation of (a) LDA, (b) mixture of unigrams, and (c) BTM. Different from LDA and mixture of unigrams, BTM models the generation procedure of biterms in a collection, rather than documents. For clarity, the fixed hyperparameters α, β are not presented.

3.1 Biterm Extraction

Without loss of generality, topics are represented as groups of correlated words in topic models, while the correlation is revealed by word co-occurrence patterns in documents. For example, if the words "apple", "iphone", "ipad" and "app" frequently co-occur with each other in the same contexts, we can identify that they belong to a same topic (i.e. apple company and its products). Conventional topic models implicitly capture such word co-occurrence patterns by modeling word generation from the document level. Different from those approaches, our BTM directly models the word co-occurrence patterns based on biterms. A biterm denotes an unordered word-pair co-occurring in a short context (i.e. an instance of word co-occurrence pattern). Here the short context refers to a proper text window containing meaningful word co-occurrences. In short texts, since documents are usually short and specific, we just take each document as an individual context unit. We extract any two distinct words in a short text document as a biterm. For example, in the short text document "I visit apple store.", if we ignoring the stop word "I", there are three biterms, i.e. "visit apple", "visit store", "apple store". The biterms extracted from all the documents in the collection compose the training data of BTM.

3.2 Biterm Topic Model

The key idea of BTM is to learn topics over short texts based on the aggregated biterms in the whole corpus to tackle the sparsity problem in single document. Specifically, we consider that the whole corpus as a mixture of topics, where each biterm is drawn from a specific topic independently[1]. The probability that a biterm drawn from a specific topic is further captured by the chances that both words in the biterm are drawn from the topic. Suppose $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the Dirichlet priors. The specific generative process of the corpus in BTM can be described as follows:

1. For each topic z
   (a) draw a topic-specific word distribution $\boldsymbol{\phi}_z \sim \text{Dir}(\boldsymbol{\beta})$
2. Draw a topic distribution $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})$ for the whole collection
3. For each biterm b in the biterm set B
   (a) draw a topic assignment $z \sim \text{Multi}(\boldsymbol{\theta})$
   (b) draw two words: $w_i, w_j \sim \text{Mulit}(\boldsymbol{\phi}_z)$

Following the above procedure, the joint probability of a biterm $b = (w_i, w_j)$ can be written as:

$$\begin{aligned} P(b) &= \sum_z P(z)P(w_i|z)P(w_j|z). \\ &= \sum_z \theta_z \phi_{i|z} \phi_{j|z} \end{aligned} \quad (1)$$

Thus the likelihood of the whole corpus is:

$$P(B) = \prod_{(i,j)} \sum_z \theta_z \phi_{i|z} \phi_{j|z} \quad (2)$$

We can see that, here we directly model the word co-occurrence pattern, rather than a single word, as an unit conveying semantics of topics. No doubt the co-occurrence of a pair of words can much better reveal the topics than the occurrence of a single word, and then enhance the learning of topics. Moreover, all the biterms from the whole corpus, rather than from a single document, are aggregated together for the topic learning. Therefore, we can fully leverage the rich global word co-occurrence patterns to better reveal the latent topics.

For better understanding the uniqueness of BTM from conventional topic models, here we make a comparison between BTM and two typical models for topic learning, i.e. LDA and mixture of unigrams. Figure 1 illustrates the graphical representation of the three models. We can see, in LDA each document is generated by first drawing a document-level topic distribution $\boldsymbol{\theta}_d$, and then iteratively sampling a topic assignment z for each word w in the document. LDA implicitly captures the document-level word co-occurrence patterns since the topic assignment variable z of each word depends on other words in the same document through sharing the same document-level topic distribution $\boldsymbol{\theta}_d$. Hence, when documents are short, LDA will suffer from the sparsity problem due to its excessive reliance on local observations for the inference of word topic assignment z, which in turn hurts the learning of topics $\boldsymbol{\phi}$.

Different from LDA, mixture of unigrams draws the topic assignment z for each document from a corpus-level topic distribution $\boldsymbol{\theta}$. Leveraging the information of the whole corpus, it alleviates the sparsity problem in topic inference,

[1]Strictly speaking, two biterms in a document sharing the same word occurrence are not independent. This simplified assumption facilitate the computation by considering BTM as a model built upon a biterm set.

which in turn helps the learning the topic components $\boldsymbol{\phi}$. However, mixture of unigrams assumes that all the words in a document are sampled from the same topic. This assumption is so strong that it prevents the model from modeling fine topics in documents. As we can see, even in short texts, there might be multiple topics in one document.

BTM, shown in Figure 1(c), overcomes the data sparsity problem of LDA by drawing topic assignment z from the corpus-level topic distribution $\boldsymbol{\theta}$ as mixture of unigrams does. Meanwhile, it also surmounts the disadvantage of mixture of unigrams by breaking documents into biterms. In this way, BTM not only can keep the correlation between words, but also can capture multiple topic gradients in a document, since the topic assignments of different biterms in a document are independent.

3.3 Inferring Topics in a Document

A major difference between BTM and conventional topic models is that BTM does not model the document generation process. Therefore, we cannot directly obtain the topic proportions of documents during the topic learning process. To infer the topics in a document, we assume that the topic proportions of a document equals to the expectation of the topic proportions of biterms generated from the document:

$$P(z|d) = \sum_b P(z|b)P(b|d). \tag{3}$$

In Eq.(3), $P(z|b)$ can be calculated via Bayes' formula based on the parameters estimated in BTM:

$$P(z|b) = \frac{P(z)P(w_i|z)P(w_j|z)}{\sum_z P(z)P(w_i|z)P(w_j|z)},$$

where $P(z) = \theta_z$, and $P(w_i|z) = \phi_{i|z}$. Then the remaining problem is how to obtain $P(b|d)$. Here we simply take the empirical distribution of biterms in the document as the estimation

$$P(b|d) = \frac{n_d(b)}{\sum_b n_d(b)},$$

where $n_d(b)$ is the frequency of the biterm b in the document d. In short texts, $P(b|d)$ is nearly an uniform distribution over all biterms in the document d. Despite of its simplicity, we find this estimation always obtains good results in practice. More sophisticated ways may be studied in the future work.

4. PARAMETERS INFERENCE

In this section, we describe the algorithm to infer the parameters $\{\boldsymbol{\phi}, \boldsymbol{\theta}\}$ in BTM, and compare its complexity with LDA.

4.1 Inference by Gibbs Sampling

Similar as LDA, inference cannot be done exactly in BTM. Hence, we adopt Gibbs sampling to perform approximate inference. Gibbs sampling is a simple and widely applicable Markov chain Monte Carlo algorithm. Compared to other inference methods for latent variable models, like variational inference and maximum posterior estimation, Gibbs sampling has two advantages. First, it is in principal more accurate since it asymptotically approaches the correct distribution. Second, it is more memory-efficient since it only requires to maintain the counters and state variables, making it preferred for large-scale dataset. More detailed comparison of these methods can be found in [1].

Algorithm 1: Gibbs sampling algorithm for BTM

Input: the number of topics K, hyperparameters α, β, biterm set B
Output: multinomial parameter $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$
initialize topic assignments randomly for all the biterms
for *iter* $= 1$ *to* N_{iter} **do**
  for $b \in B$ **do**
    draw z_b from $P(z|\mathbf{z}_{-b}, B, \alpha, \beta)$
    update n_z, $n_{w_i|z}$, and $n_{w_j|z}$
compute the parameters $\boldsymbol{\phi}$ in Eq.(5) and $\boldsymbol{\theta}$ in Eq.(6)

The basic idea of Gibbs sampling is to estimate the parameters alternatively, by replacing the value of one variable by a value drawn from the distribution of that variable conditioned on the values of the remaining variables. In BTM, we need to sample all the three types of latent variables z, $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. However, with the technique of collapsed Gibbs sampling [10], $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ can be integrated out due to the conjugate priors α and β. Consequently, we only have to sample the topic assignment for each biterm from its conditional distribution given the remaining variables.

To perform Gibbs sampling, we first choose initial states for the Markov chain randomly. Then we calculate the conditional distribution $P(z|\mathbf{z}_{-b}, B, \alpha, \beta)$ for each biterm $b = (w_i, w_j)$, where $\mathbf{z}_{-b}$ denotes the topic assignments for all biterms except b, B is the global biterm set. By applying the chain rule on the joint probability of the whole data, we can obtain the conditional probability conveniently:

$$P(z|\mathbf{z}_{-b}, B, \alpha, \beta) \propto (n_z+\alpha)\frac{(n_{w_i|z}+\beta)(n_{w_j|z}+\beta)}{(\sum_w n_{w|z}+1+M\beta)(\sum_w n_{w|z}+M\beta)}, \tag{4}$$

where n_z is the number of biterms assigned to the topic z, and $n_{w|z}$ is the number of times of the word w assigned to the topic z. Following the conventions of LDA, here we use symmetric Dirichlet priors α and β. Note that once a biterm b is assigned to the topic z, the two words w_i and w_j in it will be assigned to the topic simultaneously.

Finally, with the counters of the topic assignments of biterm and word occurrences, we can easily estimate the topic-word distributions $\boldsymbol{\phi}$ and global topic distribution $\boldsymbol{\theta}$ as:

$$\phi_{w|z} = \frac{n_{w|z}+\beta}{\sum_w n_{w|z}+M\beta}, \tag{5}$$

$$\theta_z = \frac{n_z+\alpha}{|B|+K\alpha}, \tag{6}$$

where $|B|$ is the total number of biterms.

An overview of the Gibbs sampling procedure we use is shown in Algorithm 1. Due to space limitation, we omit the detailed derivation of it.

4.2 Complexity Analysis

The major time consuming part in the Gibbs sampling procedure of BTM is evaluating the conditional probability in Eq.(4) for all the biterms, with time complexity $O(K|B|)$. During the entire process, we need to keep the counters n_z, $n_{w|z}$, and the topic assignment z for each biterm, in total of

Table 1: Time complexity and the number of variables need to be maintained in Gibbs sampling implementation of LDA, mixture of unigrams, and BTM

method	time complexity	#variables
LDA	$O(K\|D\|\bar{l})$	$\|D\|K + MK + \|D\|\bar{l}$
BTM	$O(K\|B\|)$	$K + MK + \|B\|$

Table 2: Time cost (seconds) per iteration of BTM and LDA on Tweets2011 collection.

K	50	100	150	200	250
LDA	38.07s	74.38s	108.13s	143.47s	178.66s
BTM	128.64s	250.07s	362.27s	476.19 s	591.24s

$(K+MK+|B|)$ variables in memory. Note that in LDA, we need to draw topic assignment for every word occurrence in documents, which costs time $O(K|D|\bar{l})$, where $\bar{l} = \sum_i l_i/|D|$ is the average length of documents in the collection. For memory cost, LDA has to maintain the counters $n_{z|b}$, $n_{w|z}$, and the topic assignment z for each word occurrences[15], in total of $(|D|K + MK + |D|\bar{l})$ variables. Table 1 lists the time complexity and variables required to be maintained in the Gibbs sampling procedure of LDA, and BTM.

To compare the time and memory cost between BTM and LDA, we approximately rewrite $|B|$ as[2]:

$$|B| \approx \frac{|D|\bar{l}(\bar{l}-1)}{2}.$$

We can see the time complexity of BTM is about $(\bar{l}-1)/2$ times of LDA. In short texts, the average length of documents are very small, e.g. $\bar{l} = 5.21$ in the Tweets2011 collection, thus the run-time of BTM is still comparable with LDA. However, for very large dataset and a large topic number K, LDA is susceptible to memory problems owing to a huge value of $|D|K$.

Table 2 shows the average run-time (per iteration) of BTM and LDA in our experiments on the Tweets2011 collection. We find the run-time of BTM is always about 3 times of LDA for different topic number K. Table 3 shows the overall memory cost of BTM and LDA in the same collection. We find that memory required by LDA rapidly increases as the topic number K grows, which costs more than 10 times of memory than BTM when K is larger than 200. As opposed to LDA, memory required by BTM grows very slowly. With further investigation, we find the major part of memory in BTM is used to store the biterms in training dataset. Therefore, BTM is a better choice for large dataset and a large topic number K, when the memory cost is a bottleneck.

5. EXPERIMENTS

In this section, we conduct experiments on real-world short text collections to demonstrate the effectiveness of our proposed approach. We take two typical topic models as our baseline methods, namely LDA and mixture of unigrams.

All the experiments were carried on a Linux server with Intel Xeon 2.33 GHz CPU and 16G memory. Both BTM and mixture of unigrams were implemented via C++ code[3]. For LDA, we used the open-source implementation GibbsLDA++[4]. Parameters were tuned via grid search: for LDA, $\alpha = 0.05$ and on short text collections, and $\alpha = 50/K$ on the normal text collection, $\beta = 0.01$; for BTM and mixture of unigrams, $\alpha = 50/K$ and $\beta = 0.01$. In all the methods, Gibbs sampling was run for 1,000 iterations. The results reported are the average over 10 runs.

Table 3: Memory cost (m) per iteration of BTM and LDA on Tweets2011 collection.

K	50	100	150	200	250
LDA	3177m	5524m	7890m	10218m	12561m
BTM	927m	946m	964m	984m	1002m

One typical way for topic model evaluation is to compare the perplexity or marginal likelihood on a held-out test set [3, 11, 12]. However, since BTM not models the generation process of documents, these measures are not available for us. Moreover, these measures do not reflect the topic quality rightly [6]. Therefore, we evaluate the performance of BTM on topic modeling on some other task-dependent metrics.

5.1 Evaluation on Tweets2011 Collection

To verify the effectiveness of BTM on short texts, we carried experiments on a standard short text collection, namely Tweets2011[5]. It was published in TREC 2011 microblog track, which provides approximately 16 million tweets sampled between January 23rd and February 8th, 2011. Besides the complete content of tweets, it also includes an user id, and a timestamp for each tweet. To reduce low-quality tweets, we processed the raw content via the following normalization steps: (a) removing non-Latin characters and stop words; (b) converting letters into lower case; (c) removing words with document frequency less than 10; (d) filtering out tweets with length less than 2; (e) removing duplicate tweets. At last, we left 4,230,578 valid tweets, 98,857 distinct words, and 2,039,877 users. The average document length is 5.21.

We compared BTM with three topic modeling methods on this short texts collection: (a) the standard LDA, which takes each tweet as a document; (b) LDA-U, which aggregates all the tweets from a user to a big psudo-document before training LDA; (c) mixture of unigrams (denoted as Mix), which assumes each tweet only exhibits a single topic. In this collection, we set the number of topics $K = 50$ for all the methods.

5.1.1 *Quality of Topics*

To investigate the quality of topics discovered by all the test methods, we first sample some topics for visualization. Following [7], we randomly drew two topics shared by the topic sets discovered by the four methods. The selection process is described as follows. Firstly, we collected the top 5 words in each topic into a topical word set for each method individually. Then we randomly chose two terms (i.e., job and snow) from the intersection of the four topical word sets. For each topic, besides the top 20 words, which are

[2]For a document with length l, we generate $l(l-1)/2$ biterms. Here we simply take all the documents as with the same length, since the variance of the length of short documents is not large.

[3]Code of BTM : http://code.google.com/p/btm/

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

[5]http://trec.nist.gov/data/tweets/

most representative for a topic, we also listed 20 non-top words (i.e. ranked from 1001 to 1020) ordered by $P(w|z)$. Ideally, a high quality topic should be coherent as much as possible. Hence, it is expected that the non-top words should be relevant to the top words in the same topic.

Table 4 presents the top words (first row) and non-top words (second row) of the topic selected by the word "job". We find the two words "job" and "jobs" are ranked highest by all the four methods. However, in LDA, some other words, like "web", "website", and "google", are more related to a topic about website, rather than job. The results in LDA-U and mixture of unigrams seem a little better than LDA, but still include a few of less relevant words like "website" and "www". While in BTM, the top 20 words are more prominent and precise about "job". In the non-top words, we find LDA includes the least words about "job", which is hard to connect them to the top words. On the contrary, BTM includes more relevant words about "job" than others, suggesting this topic discovered by BTM is more coherent.

Table 5 presents the top words (first row) and non-top words (second row) of the topic selected by another word "snow". In the first row, again we can see that the top words in LDA are mixed with words about two different subjects "weather" and "car". The results in LDA-U is similar to LDA, but more about "weather". In contrast, the top words in mixture of unigrams and BTM clearly describe weather. In the second row, both LDA and LDA-U list words almost have no connection to "snow", while some of them are related to "car". For mixture of unigrams, it is hard to explain the topic based on these non-top words. In BTM, there are still many words about "weather", like "temperature" and "cyclone". Besides the two topics presented here, we also find similar phenomenon in remaining topics, which suggests that the topics discovered by BTM are is more prominent and coherent than the three baselines.

In order to perform more comprehensive analysis, we utilize an automated metric, namely *coherence score*, proposed by Mimno et al [21] for topic quality evaluation. Given a topic z and its top T words $V^{(z)} = (v_1^{(z)}, ..., v_T^{(z)})$ ordered by $P(w|z)$, the coherence score is defined as:

$$C(z; V^{(z)}) = \sum_{t=2}^{T} \sum_{l=1}^{t} \log \frac{D(v_t^{(z)}, v_l^{(z)}) + 1}{D(v_l^{(z)})},$$

where $D(v)$ is the document frequency of word v, $D(v, v')$ is the number of documents words v and v' co-occurred. The coherence score is based on the idea that words belonging to a single concept will tend to co-occur within the same documents. It is empirically demonstrated that the coherence score is highly correlated with human-judged topic coherence. It must be stressed that the coherence score only is appropriate for measuring frequent words in a topic. Because the frequency of rare words is less reliable.

To evaluate the overall quality of a topic set, we calculated the average coherence score, namely $\frac{1}{K}\sum_k C(z_k; V^{(z_k)})$, for each method. The result is listed in Table 6, where the number of top words T ranges from 5 to 20. We find the result is in agreement with previous qualitative analysis. BTM receives the highest coherence score in all the settings, and the superiority is statistically significant (P-value < 0.01 by T-test). Both LDA-U and mixture of unigrams outperform LDA slightly, but the differences are not significant.

Table 6: Average coherence score on the top T words (ordered by $P(w|z)$) in topics discovered by LDA, LDA-U, mixture of unigrams, and BTM. A larger coherence score means the topics are more coherent. It suggests that BTM outperforms others significantly (P-value < 0.01 by t-test).

T	5	10	20
LDA	-55.0 ± 0.4	-236.4 ± 2.0	-1015.7 ± 5.9
LDA-U	-54.2 ± 0.8	-234.8 ± 1.1	-1009.4 ± 4.4
Mix	-53.8 ± 0.1	-233.0 ± 1.4	-1007.6 ± 6.7
BTM	$\mathbf{-52.4 \pm 0.1}$	$\mathbf{-227.8 \pm 0.3}$	$\mathbf{-990.2 \pm 3.8}$

Table 7: Hashtags used for evaluation, not including the prefix '#'.

jan25 superbowl sotu wheniwaslittle mobsterworld jobs
agoodboyfriend bieberfact glee lfc rhoa itunes thegame
celebrity tcyasi americanidol cancer socialmedia jerseyshore
photography jp6foot7remix factsaboutboys meatschool
libra android sagittarius thissummer tnfisherman sagawards
ausopen bears weather jaejoongday skins bfgw fashion
pandora realestate teamautism travel nba football marketing
design oscars food dating kindle snow obama

5.1.2 Quality of Topical Representation of Documents

In the Tweets2011 collection, there is no category information for tweets. Manual labeling might be difficult due to the incomplete and informal content of tweets. Fortunately, some tweets are labeled by their authors with hashtags in the form of "#keyword". By investigating the data, we find there are mainly three types of usage of hashtags: (a) marking events or topics; (b) defining the types of content, like "#ijustsayin", "#quote"; (c) realizing some specified functions, like "#fb" means importing the tweet to Facebook in the meanwhile. In our case, only the first type of hashtags are useful. Therefore, we manually chose 50 frequent hashtags in type (a), listed in Table 7.

Since each hashtag in Table 7 denotes a specific topic labeled by its author, we organized documents with the same hashtag into a cluster. The following evaluation is based on the fact that these clusters should have low intra-cluster distances and high inter-cluster distances.

Considering topic models as a type of dimension reduction methods, each document can be represented by a vector of posterior distribution of topics:

$$d_i = [p(z_1|d_i), ..., p(z_k|d_i)]. \quad (7)$$

Then we can measure the distance of two documents by the Jensen–Shannon divergence:

$$dis(d_i, d_j) = \frac{1}{2} D_{KL}(d_i||m) + \frac{1}{2} D_{KL}(d_j||m),$$

where $m = \frac{1}{2}(d_i + d_j)$, and $D_{KL}(p||q) = \sum_i p_i \ln \frac{p_i}{q_i}$ is the Kullback–Leibler divergence. Given a set of clusters $C = \{C_1, ..., C_K\}$, we introduce two distance scores

Average Intra-Cluster Distance:

$$IntraDis(C) = \frac{1}{K} \sum_{k=1}^{K} \left[\sum_{\substack{d_i, d_j \in C_k \\ i \neq j}} \frac{2dis(d_i, d_j)}{|C_k||C_k - 1|} \right]$$

Table 4: Topics selected by the word "job" on the Tweets collection. The first row lists the top 20 words, while the second row lists non-top words ranked from 1001 to 1020 based on $P(w|z)$.

LDA	LDA-U	Mixture of unigrams	BTM
job jobs business web	**job** jobs design manager	jobs **job** business	jobs **job** manager business
website google design online	project web website site	marketing social media	sales hiring service services
marketing site blog project	business service	online web design website	project company senior
manager search	company hiring www	manager blog project seo	engineer management
www company service	support sales services	internet sales tips	marketing nurse office assistant
sales services post	london blog senior engineer	company site hiring	center customer development
nonprofit gallery announced	expertise unemployed med iii	understand rep industrial	springfield mlm recruit oil req
presence published converting	host educational fort tags	sustainability rankings	unemployment processing
select reps requirement mgr	apps assignments labor	scholarships stay single campus	overview awards recruiters
territory recruiters power	introduction leads github	extra cheap 101 vp relationships	ict finish entrepreneur comp
involved announce poster	assurance avon manchester	beginners colorado compliance	assist 1000 alliance locations
larry dynamics feeds bristol	starting automotive table	face winning mechanical	patent auditor

Table 5: Topics selected by the word "snow" on the Tweets collection. The first row lists the top 20 words, while the second row lists non-top words ranked from 1001 to 1020 based on $P(w|z)$.

LDA	LDA-U	Mixture of unigrams	BTM
snow car weather cold	**snow** weather cold winter	**snow** weather cold storm	**snow** cold weather early
drive storm winter ice	ice storm rain stay	winter ice rain warm	stay ready ice winter
road bus driving rain	warm due car closed	degrees stay sun spring	storm hour hours weekend
ride traffic cars safe	coming spring drive traffic	safe blizzard coming wind	warm late coming spring
closed due warm train	safe sun blizzard city	cyclone chicago freezing inches	rain tired sun hot
western dmv covering a4	locations sunset drizzle	australian thankful station	temperature cyclone
push pulling milwaukee	mississippi interstate residents	stops groundhogday possibly	warmth issued colder
remains pace idiots 95	portland students fireplace	cleveland traveling sidewalk	mood couch snows pre
commuter buick owner	letting yuck ton counties signal	covering predicting ten grass	traveling polar outages
cta transmission cyclist	counting blankets pushed	meant double affect	umbrella filled yawn outage
flurries camping tyre	3pm springfield venture	zoo schedule blew causing	flurries online gloves speed

Average Inter-Cluster Distance:

$$InterDis(C) = \frac{1}{K(K-1)} \sum_{\substack{C_k, C_{k'} \in C \\ k \neq k'}} \left[\sum_{d_i \in C_k} \sum_{d_j \in C_{k'}} \frac{dis(d_i, d_j)}{|C_k||C_{k'}|} \right]$$

The intuition is that if the average inter-cluster distance is small compared to the average intra-cluster distance, the topical representation of documents agrees well with human labeled clusters (via hashtag). Therefore, we calculate the following ratio to evaluate the quality of one topical representation of documents as [4, 13]:

$$H = \frac{IntraDis(C)}{InterDis(C)}.$$

Given a set of different topical representations of documents, the best one is which minimizes the H score.

Table 8 shows the H score for all the test methods. From the results, we can see that BTM preforms significantly better than other three methods (P-value < 0.001). LDA-U outperforms LDA slightly, implying that aggregating tweets for individual users brings moderate benefit. Although LDA dominates mixture of unigrams on normal texts, it is somehow surprising that the performance of mixture of unigrams outperforms LDA and LDA-U substantially in this short text collection. It suggests that the data sparsity problem seriously affects LDA and LDA-U, while less influences mixture of unigrams and BTM. However, the H score of mixture of ungirams is still much worse than BTM. With some further analysis, we find the average intra-cluster distance of mixture of unigrams is extremely large, owing to its peaked posterior distribution of $P(z|d)$. In other words, mixture of unigrams fails to recognize the resemblance of many documents.

Table 8: H score for different methods on the Tweets2011 collection, smaller value is better. The significant levels(P-value by t-test) are denoted as 0.1*, 0.01, 0.001***.**

Method	H score	Significant differences
LDA	0.576 ± 0.007	
LDA-U	0.564 ± 0.011	>LDA*
Mix	0.503 ± 0.008	>LDA-U**>LDA***
BTM	0.474 ± 0.005	>Mix***>LDA-U***>LDA***

From the above results, we find the improvement of LDA-U over LDA is not so much as shown in [17]. An explanation for this difference is that there are less tweets posted by an user in average in our dataset than theirs. Figure 2 shows the proportions of users who posted certain number of tweets in the Tweets2011 collection, we find 63.3% of users posted one tweet, and only 2.1% of users posted more than 9 tweets. Thus it is not strange that aggregating tweets for individual users has limited affects.

5.2 Evaluation on Question Collection

In order to demonstrate the effectiveness of our approach is domain-independent, we evaluated it on another short text collection, called Question collection. This collection includes 648,514 questions crawled from a popular Chinese Q&A website[6]. Each question has a category label assigned by its questioner, making it convenient for automatic evalu-

[6]http://zhidao.baidu.com

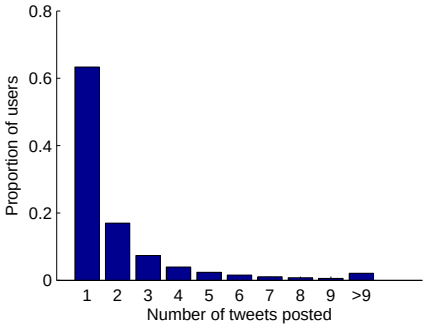

Figure 2: Proportions of users who posted certain number of tweets in the Tweets2011 collection.

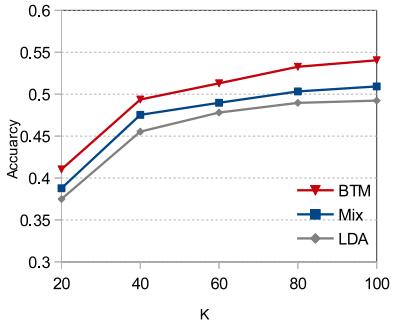

Figure 3: Classification performance of BTM, mixture of unigrams, and LDA on the Question collection.

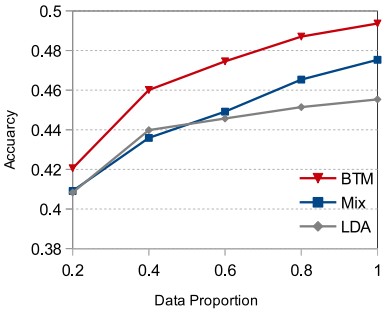

Figure 4: Classification performance comparison with different data proportions on the Questions collection (K=40).

ation. For pre-process, we removed stop words and low frequency words (i.e. document frequency is less than 3). The final collection contains 189,080 documents, 26,565 distinct words, and 35 categories. The average length of documents is 3.94. Note that in this collection, our baselines do not include LDA-U, since there is few users whole submitted more than one question.

We performed the evaluation based on document classification. Considering topic model as a way for dimensionality reduction, which reduces a document to a fixed set of topical features $P(z|d)$, we would like to see how accurate and discriminative of the topical representation of documents for classification. We randomly split documents into training and test subsets with the ratio 4 : 1, and classified them by the linear SVM classifier LIBLINEAR[7]. We reported the accuracy on 5-fold cross validation in Figure 3.

From the results, we can see that BTM always dominates the two baselines. Moreover, the advantage of BTM becomes more notable as the topic number K grows. That is because when the number of topics is small, topics discovered are usually very general. In such case, a short document is more likely to belong to a single topic, thus the performance of BTM is close to mixture of unigrams. In contrast, with the increase of the topic number K, we will learn more specific topics. However, mixture of unigrams is unable to capture the multiple topics exhibited in a document. Thus the difference between BTM and mixture of unigrams becomes larger. At the same time, a large topic number will aggravate the data sparsity problem of LDA by introducing more parameters, thus the gap between BTM and LDA also increases. Another important finding is that mixture of unigrams outperforms LDA all the time. It suggests that LDA is not a good choice for short texts due to the data sparsity problem.

One may wonder the impact of training data size on these methods. We randomly sampled different proportion of documents, from 0.2 to 1, to train and test these methods separately. The results are shown in Figure 4. We can see when the size of the training data grows, all the methods work better. However, both BTM and mixture of unigrams achieve more improvement than LDA. LDA only get close to mixture of unigrams on small training data. It suggests that increasing the training data will not overcome the data sparsity problem in LDA, since the documents are still short.

[7]http://www.csie.ntu.edu.tw/~cjlin/liblinear/

Comparing mixture of unigrams with BTM, we find BTM has stable superiority over mixture of unigrams no matter of the size of the training data.

5.3 Evaluation on Normal Texts

In previous experiments, we have demonstrated the effectiveness of BTM on short texts. Although we propose BTM for the short text scenario, there is no limitation for our model to be applied on normal text collections. Therefore, it is also interesting to see how effective is BTM on normal text. For this purpose, we compared BTM with LDA, one of most popular topic models, on a normal text collection. The experiments were carried out on the 20Newsgroup collection[8], a standard corpora including 18,828 messages harvested from 20 different Usenet newsgroups. Each newsgroup corresponding to a different topic. Table 9 lists the names of these newsgroups. For pre-process, we removed stop words and words with frequency less than 3, but without stemming. Finally, 42697 words are left.

We directly trained LDA on the original documents without any other processing. Note that in BTM, we need to extract biterms from the collection. This process is a little different from that in short texts. Recall that a biterm is defined as a word-pair co-occurred in a short context. It is not appropriate to view a lengthy document as a single short context, since it may involve a wide range of topics. In or-

[8]http://qwone.com/~jason/20Newsgroups/

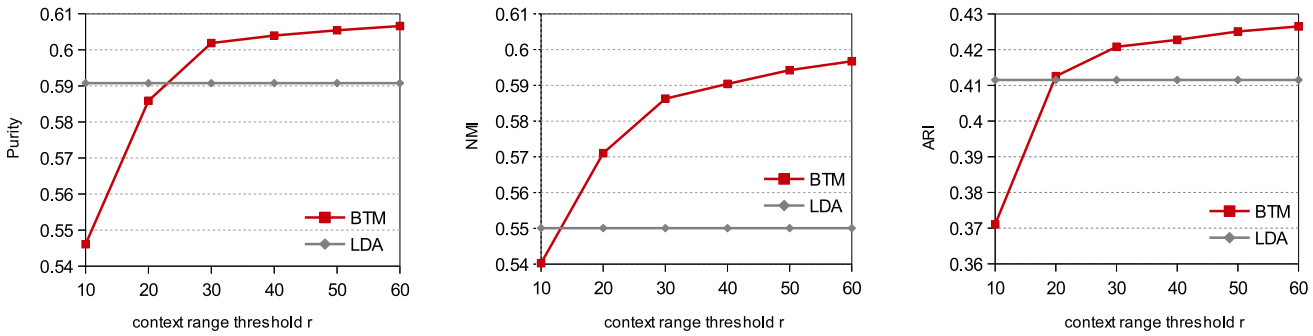

Figure 5: Clustering performance of BTM with different context range thresholds and LDA on the 20 Newsgroups collection ($K = 20$).

Table 9: The newsgroup names in the 20 Newsgroups collection

No.	Newsgroup Name	No.	Newsgroup Name
1	alt.atheism	11	rec.sport.hockey
2	comp.graphics	12	sci.crypt
3	comp.os.ms-windows.misc	13	sci.electronics
4	comp.sys.ibm.pc.hardware	14	sci.med
5	comp.sys.mac.hardware	15	sci.space
6	comp.windows.x	16	soc.religion.christian
7	misc.forsale	17	talk.politics.guns
8	rec.autos	18	talk.politics.mideast
9	rec.motorcycles	19	talk.politics.misc
10	rec.sport.baseball	20	talk.religion.misc

der to reduce meaningless and noise biterms, the biterm set is constructed by extracting any two words co-occur within a context window with range no larger than a predefined threshold r in each document.

5.3.1 *Quantitative Evaluation*

For quantitative evaluation, we compare the clustering performance of BTM and LDA. Document clustering evaluation is a direct way to measure the effectiveness of a topic model without depending on any extrinsic methods. For document clustering, we take each topic as a cluster, and assign each document d to the topic z with highest value of conditional probability $P(z|d)$. Note that we do not know the optimal context range threshold r ahead, therefore, we tested different values of it, and report their results together.

We adopt three standard metrics in clustering evaluation as follows. Let $\Omega = \{\omega_1, \cdots, \omega_K\}$ be the set of output clusters, and $\mathbb{C} = \{c_1, \cdots, c_P\}$ be P labeled classes of the documents.

- Purity. Suppose documents in each cluster should take the dominant class in the cluster. Purity is the accuracy of this assignment measured by counting the number of correctly assigned documents and divides by the total number of test documents. Formally:

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{n}\sum_{i=1}^{K}\max_j |\omega_i \cap c_j|.$$

  Note that when all the documents in each cluster are with the same class, purity is highest with value of 1. Conversely, it is close to 0 for bad clustering.

- Normalized Mutual Information(NMI). Let $I(\Omega;\mathbb{C})$ denotes the mutual information between the two partitions Ω and $\mathbb{C}$, NMI penalized $I(\Omega;\mathbb{C})$ by their entropy $H(\Omega)$ and $H(\mathbb{C})$ to avoid the value biasing to large number of clusters. Formally:

$$\begin{aligned}\text{NMI}(\Omega,\mathbb{C}) &= \frac{I(\Omega;\mathbb{C})}{[H(\Omega)+H(\mathbb{C})]/2} \\ &= \frac{\sum_{i,j}\frac{|\omega_i\cap c_j|}{n}\log\frac{|\omega_i||c_j|}{n|\omega_i\cap c_j|}}{(\sum_i\frac{|\omega_i|}{n}\log\frac{|\omega_i|}{n}+\sum_j\frac{|c_j|}{n}\log\frac{|c_j|}{n})/2}\end{aligned}$$

  Note that NMI is 1 for perfect match between Ω and $\mathbb{C}$, while 0 if the clustering is random with respect to class membership.

- Adjusted Rand Index(ARI)[18]. Consider documents clustering as a series of pair-wise decisions. If two documents both in the same class and the same cluster, or both in different classes and different clusters, the decision is considered to be correct, else false. Rand index measures the percentage of decisions that are correct. Adjusted Rand index is the corrected-for-chance version of Rand index, whose expected value is 0, while the maximum value is also 1 for exactly match.

$$ARI = \frac{\sum_{i,j}\binom{|\omega_i\cap c_j|}{2} - [\sum_i\binom{|\omega_i|}{2}\sum_j\binom{|c_j|}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i\binom{|\omega_i|}{2}+\sum_j\binom{|c_j|}{2}] - [\sum_i\binom{|\omega_i|}{2}\sum_j\binom{|c_j|}{2}]/\binom{n}{2}}$$

The results are shown in Figure 5. On the whole, it is clear that BTM outperforms LDA significantly when the context range threshold r is between 30 and 60, suggesting that BTM also performs very well on normal texts. In particular, we find when $r = 10$, LDA works better than BTM, implying that the context information utilized by BTM is not enough. As the context range threshold r increases, more word co-occurrence patterns are included, which improves the performance of BTM substantially. However, the improvement slows down when the context range threshold r increases from 30 to 60. An explanation for this behavior is that when the distance between two words increasing, they might be less relevant. At this point, the assumption that the two words in a biterm have the same topic will be less credible. Moreover, a larger context range threshold r will generate much more biterms, which increases the training cost. Therefore, for both effectiveness and efficiency consideration, the context range threshold r should not be too small or too large for normal texts in practice.

Table 10: Topics discovered from the 20 Newsgroup collection by BTM and LDA (K=20). "sim" in the last column denotes the cosine similarity of the two topics in a row.

	BTM	LDA	sim
1	ax max g9v b8f a86 1d9 pl 145 3t giz	ax max b8f g9v a86 145 1d9 pl 0t 3t	0.99
2	god jesus christ church bible people lord christian	god jesus bible christian church christ christians paul	0.95
3	key encryption chip clipper keys government system	key encryption chip clipper government keys public	0.95
4	window server display widget set application xterm file	window server set application sun display problem manager	0.93
5	space earth launch mission orbit shuttle system solar	space earth nasa gov time system mission launch	0.91
6	writes article don ca david uk wrote cs org	writes article university uk ca cs michael mail brian	0.90
7	ax 0d cx 145 ah 34u w7 mv scx uw	0d cx ah w7 mv sp 17 uw scx air	0.86
8	people don fbi fire children koresh gun batf	people writes gun fbi fire children article koresh	0.83
9	people don god writes make good point question	people writes true don religion evidence question god	0.82
10	people government president don make time american	president government people state states rights american	0.80
11	disease medical people patients don time writes good	medical health disease drug study drugs men cancer	0.79
12	drive scsi mac bit card apple system monitor problem	windows drive dos card mac system apple scsi disk	0.75
13	image jpeg file graphics images files color data format	file image program files bit jpeg color output line	0.74
14	mail university information fax internet list email	graphics ftp software data mail pub computer	0.62
15	car don writes cars good ve engine time	car cars armenian armenians engine muslims turkish 000	0.62
16	00 year team 10 game 55 play players games 20	writes year play game good ca insurance scott team games	0.61
17	1993 health men number 10 hiv april study homosexual	10 1993 20 15 00 12 93 11 30	0.54
18	windows dos file system files run don os pc program	don people ve time good ll make things thing doesn	0.25
19	armenian armenians people war muslims turkish	information group list book post questions read subject	0.15
20	file entry output program build line printf char info	writes price buy sale problem cost power good interested	0.03

5.3.2 *Qualitative Evaluation*

Here we study the quality of topics discovered by the two topic models. In practice, a topic model which finds topics with good readability and accurately reflecting the topical structure of data is preferred. Table 10 presents all the topics learned by BTM and LDA, when the number of topics is set to 20. These topics from the two methods are matched based cosine similarity using greedy algorithm. For each topic we list its top words ordered by $P(w|z)$. We can see that the topics 1-16 in BTM and LDA are very similar. Comparison Table 9 and Table 10, we find it is easy to identify the corresponding newsgroup of a topic in topics 1-16, except topic 1 and topic 7. For example, topic 2 is with respect to the newsgroup "soc.religion.christian". It suggests that both BTM and LDA uncover the inherent topical structure of the collection closely.

We also note that topics 17-20 in Table 10 are very different in BTM and LDA. In BTM, we can still identify that topics 17-20 relate to the newsgroups "sci.med", "comp.os.ms-windows.misc", "talk.politics.mideast", "comp.os.ms-windows.misc" respectively. But in LDA, topic 17 is about numeral, topic 18 is a set of common words, while topics 19 and 20 are with poor interpretability. In our view, the differences between the results of the two models are caused by the following reasons. BTM explicitly model the word co-occurrences in local context, it well captures the short-range dependencies between words. Conversely, LDA captures the long-range dependencies in documents [11], which are less specific than short-range ones, resulting in the last four topics more common but less readable.

6. CONCLUSION & FUTURE WORKS

Topic modeling for short texts is an increasingly important task due to the prevalence of short texts on the Web. Compared to normal documents, short texts lack of word frequency and context information, causing severe sparsity problems for conventional topic models. In this paper, we propose a novel probabilistic topic model for short texts, namely biterm topic model (BTM). BTM can well capture the topics within short texts as it explicitly models the word co-occurrence patterns and uses the aggregated patterns in the whole corpus. We carried on experiments on two real-world short text collections and one normal text collection. The results demonstrated that BTM not only can learn higher quality topics, but also more accurately capture the topics of documents than previous methods. Besides, BTM is simple and easy to implement, and also scales up well. All these benefits makes BTM a practicable choice for content analysis on short texts in a wide range of applications.

To the best of our knowledge, we are the first to propose a topic model for general short texts. However, there is still room to improve our work in the future. For example, we would like to find more sophisticated way to estimate the distribution $P(b|d)$, which is uniform in the current work for simplicity. Moreover, it is also interesting to explore the usage of our model in various real-world applications, like content recommendation, event tracking, and short texts retrieval, etc.

7. ACKNOWLEDGEMENTS

This work is funded by the National Natural Science Foundation of China under Grant No. 61202213, 61203298, No. 60933005, No. 61173008, No. 61003166, and 973 Program of China under Grants No. 2012CB316303. We would like to thank the anonymous reviewers for their helpful comments.

8. REFERENCES

[1] A. Asuncion, M. Welling, P. Smyth, and Y. Teh. On smoothing and inference for topic models. In *In Proceedings of the 25th Conference on UAI*, 2009.

[2] D. Blei and J. McAuliffe. Supervised topic models. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. MIT Press, Cambridge, MA, 2008.

[3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] I. Bordino, C. Castillo, D. Donato, and A. Gionis. Query similarity by projecting the query-flow graph. In *SIGIR*, pages 515–522. ACM, 2010.

[5] J. Boyd-Graber and D. M. Blei. Syntactic topic models. Technical Report arXiv:1002.4665, Feb 2010.

[6] J. Boyd-Graber, J. Chang, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

[7] D. Cai, Q. Mei, J. Han, and C. Zhai. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 911–920. ACM, 2008.

[8] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1185–1194. ACM, 2010.

[9] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[10] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[11] T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. Integrating topics and syntax. *NIPS*, 17:537–544, 2005.

[12] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Hidden topic markov models. *Artificial Intelligence and Statistics (AISTATS)*, 2007.

[13] J. Guo, X. Cheng, G. Xu, and X. Zhu. Intent-aware query similarity. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 259–268. ACM, 2011.

[14] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *SIGIR*, pages 267–274. ACM, 2009.

[15] G. Heinrich. Parameter estimation for text analysis. *Technical report*, 2005.

[16] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57. ACM, 1999.

[17] L. Hong and B. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[18] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[19] O. Jin, N. Liu, K. Zhao, Y. Yu, and Q. Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 775–784. ACM, 2011.

[20] C. X. Lin, B. Zhao, Q. Mei, and J. Han. Pet: a statistical model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD*, pages 929–938. ACM, 2010.

[21] D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics, 2011.

[22] D. Newman, E. V. Bonilla, and W. Buntine. Improving topic coherence with regularized topic models. In *Advances in Neural Information Processing Systems 24*, pages 496–504. 2011.

[23] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2):103–134, 2000.

[24] X. Phan, L. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100. ACM, 2008.

[25] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388, New York, NY, USA, 2009. ACM.

[26] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *International AAAI Conference on Weblogs and Social Media*, volume 5, pages 130–137, 2010.

[27] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[28] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, pages 377–386. ACM, 2006.

[29] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[30] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD*, pages 424–433, New York, NY, USA, 2006. ACM.

[31] Y. Wang, E. Agichtein, and M. Benzi. Tm-lda: efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD*, pages 123–131, New York, NY, USA, 2012. ACM.

[32] J. Weng, E. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[33] X. Yan, J. Guo, S. Liu, X. Cheng, and Y. Wang. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 2013.

[34] X. Yan, J. Guo, S. Liu, X.-q. Cheng, and Y. Wang. Clustering short text using ncut-weighted non-negative matrix factorization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2259–2262, New York, NY, USA, 2012. ACM.

[35] W. Zhao, J. Jiang, J. Weng, J. He, E. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. *Advances in Information Retrieval*, pages 338–349, 2011.