


# A topic model for co-occurring normal documents and short texts

Yang Yang<sup>1</sup> · Feifei Wang<sup>2</sup>  · Junni Zhang<sup>3</sup> · Jin Xu<sup>1</sup> · Philip S. Yu<sup>4</sup>

Received: 19 September 2016 / Revised: 28 February 2017 /  
Accepted: 8 May 2017 / Published online: 23 June 2017  
© Springer Science+Business Media New York 2017

**Abstract** User comments, as a large group of online short texts, are becoming increasingly prevalent with the development of online communications. These short texts are characterized by their co-occurrences with usually lengthier normal documents. For example, there could be multiple user comments following one news article, or multiple reader reviews following one blog post. The co-occurring structure inherent in such text corpora is important for efficient learning of topics, but is rarely captured by conventional topic models. To capture such structure, we propose a topic model for co-occurring documents, referred to as COTM. In COTM, we assume there are two sets of topics: formal topics and informal topics, where formal topics can appear in both normal documents and short texts whereas informal topics can only appear in short texts. Each normal document has a probability distribution over a set of formal topics; each short text is composed of two topics, one from the set of formal topics, whose selection is governed by the topic probabilities of the

---

✉ Feifei Wang  
wangff@pku.edu.cn

Yang Yang  
yyang1988@pku.edu.cn

Junni Zhang  
zjn@gsm.pku.edu.cn

Jin Xu  
jxu@pku.edu.cn

Philip S. Yu  
psyu@cs.uic.edu

<sup>1</sup> School of Electrical Engineering and Computer Science, Peking University, Beijing, China

<sup>2</sup> School of Statistics, Renmin University of China, Beijing, China

<sup>3</sup> Guanghua School of Management, Peking University, Beijing, China

<sup>4</sup> Department of Computer Science, University of Illinois at Chicago, Chicago, USA

corresponding normal document, and the other from a set of informal topics. We also develop an online algorithm for COTM to deal with large scale corpus. Extensive experiments on real-world datasets demonstrate that COTM and its online algorithm outperform state-of-art methods by discovering more prominent, coherent and comprehensive topics.

**Keywords** Co-occurring structure · Online algorithm · Short texts · Topic model

## 1 Introduction

With more online service providers encouraging users to make comments or leave feedbacks to their real-time updated contents, co-occurring normal documents and short texts are constantly generated throughout the Internet. For example, each news article in news publishing platforms could be followed by multiple reader comments, each blog post in blog websites could be followed by multiple reader reviews, and each product description in electronic commerce websites could be followed by multiple consumer reviews. The short texts may discuss issues addressed in their corresponding normal documents, and may also discuss other issues, such as personal opinions. The co-occurring structure inherent in such text corpora poses challenges to conventional topic modeling.

Topic models [2, 9] have been successfully applied to modeling normal documents, such as news articles, blog posts and product descriptions, and have achieved great success in uncovering latent semantic structure. In the basic Latent Dirichlet Allocation (LDA) model [2], documents are taken as mixtures of topics and each topic has a probability distribution over a dictionary of words. It has also been extended in various ways to deal with more complicated modeling tasks. For example, Liu et al. [15] propose a model that jointly models the generation of contents and friendships of authors in social networks, within which the user topics and the link formation pattern can be learned in a unified model. McCallum et al. [17] propose a model to simultaneously discover groups among the entities and topics among the corresponding texts. Nagarajan et al. [21] propose a probabilistic model for community structures and user contents that can discover coherent communities and topics at the same time.

When faced with a corpus of short texts, LDA and its extensions suffer from severe data sparsity problem. Specifically, 1) small word counts in short texts restrict the ability of topic models to learn how words are related, and hence the learnt topics are less discriminative than those learnt from normal documents [10]; 2) limited contexts in short texts make it more difficult for topic models to distinguish ambiguous words [27].

Two major heuristic strategies have been adopted to alleviate this sparsity problem. The first strategy aggregates short texts into pseudo-documents. It is widely used in social media but is highly data-dependent. For example, Weng et al. [26] aggregate tweets belonging to the same user, Hong et al. [10] aggregate tweets containing the same word and Mehrotra et al. [18] aggregate tweets based on hashtags. Conventional topic models are then applied to the pseudo-documents to learn more prominent topics from the enriched contexts of aggregated texts. However, auxiliary information such as authorship or hashtag is not always available in real word applications. Another strategy is to extend topic models by adding strong assumptions on short texts. Zhao et al. [29] and Lakkaraju et al. [13] assume each short text is a mixture of unigrams sampled from only one topic. The biterm topic model (BTM) [5, 27] turns the whole corpus into a biterm set, where a biterm is constructed by any two distinct words in a short context. BTM then assumes that the two words in any biterm are drawn independently from a topic, where the topic is sampled from a topic mixture

over the whole corpus. The self-aggregation topic model [25] assumes each piece of short text is sampled from unobserved pseudo-documents and automatically aggregates short texts. Also using the self-aggregation method, Zuo et al. [30] propose a Pseudo-document-based Topic Model (PTM) for short texts, which can solve the overfitting problem and save computational cost in [25].

Recently, word embedding models [11, 19, 22] have gained much attention with their ability to form clusters of conceptually similar words in the embedding space. [11] proposes a latent concept topic model (LCTM), which models each topic as a distribution over the latent concepts and each concept is a Gaussian distribution over the word embedding space. Since the number of concepts is often much smaller than the number of unique words, LCTM is less susceptible to the data sparsity.

The methods which explore external normal texts to improve topic learning of short texts are closely related to our work. For example, Phan et al. [23, 24] propose to train topic models on a collection of long texts which are in the same domain as the short texts, and then make inference on the short texts to help the learning of their topics. Jin et al. [12] learned topics on short texts via transferring knowledge from auxiliary long text data. Performance of these approaches, however, is highly data-dependent, as the quality of topic learning is highly dependent on the quality of the organization of external datasets. Targeted on summarization of short texts, Ma et al. [16] utilize the relationships between normal documents and corresponding short texts to enhance topic learning of short texts. They propose two models, the Master-Slave Topic Model (MSTM) to restrict topics of short texts within those of their associated normal documents, and the Extended Master-Slave Topic Model (ESTM) to allow some short texts to represent topics only extracted from themselves and not correlated with normal documents. However, both MSTM and ESTM miss the situation that short texts may not only contain content information from their associated normal documents and also express their own opinions.

In this paper, we fill this gap and propose a co-occurring topic model COTM, which can directly exploit the co-occurring structure in the text corpora and utilize information from both the normal documents and the short texts for efficient topic learning. We assume (1) each normal document has a probability distribution over a set of formal topics; (2) each short text has a probability distribution over two topics, one belonging to the formal topics, whose selection is governed by the topic probabilities of the corresponding normal document, and the other belonging to a set of informal topics which are shared only by short texts. Intuitively, for each short text, its formal topic is the one that appeals to its author from the set of topics for the corresponding normal document, and its informal topic reflects the additional discussion that its author adds to the chosen formal topic. As a result, in COTM, topic modeling of normal documents is enhanced by the inclusion of words from the corresponding short texts that are relevant to the formal topics, and the informal topics are learnt from words that are irrelevant to formal topics but shared across short texts.

In practice, texts are constantly generated, and online scalable inference algorithms are needed. For co-occurring text corpora, short texts can be created at any timestamps after the corresponding normal documents are published. We introduce an online algorithm for COTM, referred to as oCOTM, to deal with dynamically generated co-occurring documents. The oCOTM algorithm can incrementally adjust the learned topics according to the dynamic stream of data, without need to access previously processed texts. Compared with COTM, the advantage of oCOTM is that it only needs to store a small fraction of data for model update, saving both computational cost and memory cache.

We conduct extensive experiments on two large real-world text collections, i.e. news articles together with reader comments from NetEase news website, and blog posts together

with user comments from Sina blog website. Experiments on both batch and online algorithms show that (1) COTM learns more coherent and comprehensive topics than several state-of-art methods for topic modeling, like LDA, BTM and EXTM; (2) the topic proportions obtained by COTM can better help document clustering and classification, indicating that COTM offers better topic representations than its competitors. Moreover, COTM properly reveals topical relationships between normal documents and their ensuing short texts, which can be effectively used in detecting spam user comments.

This paper extends our previous conference article [28] with the following improvements: 1) we introduce an online algorithm for COTM to handle continuously generated texts, including both short texts and normal documents. 2) Both batch and online COTM algorithms are empirically verified with more comprehensive experiments. The rest of this paper is organized as follows. Sections 2 and 3 present the batch and online implementations of COTM. Section 4 shows experimental results. Section 5 then concludes.

## 2 The COTM model

Co-occurring documents, consisting of both normal documents and short texts, are illustrated in Figure 1. Borrowing ideas from previous works, we use normal documents as auxiliary information to help improve the topic learning for short texts. On the other hand, we also enhance topic learning for normal documents by using information from the corresponding short texts.

### 2.1 Model description

We assume the generative process of normal documents follow the LDA model. Assume that there are  $K$  topics underlying  $D$  normal documents, which we refer to as formal topics thereafter. Each normal document  $d$  is a mixture of the  $K$  formal topics with its own vector

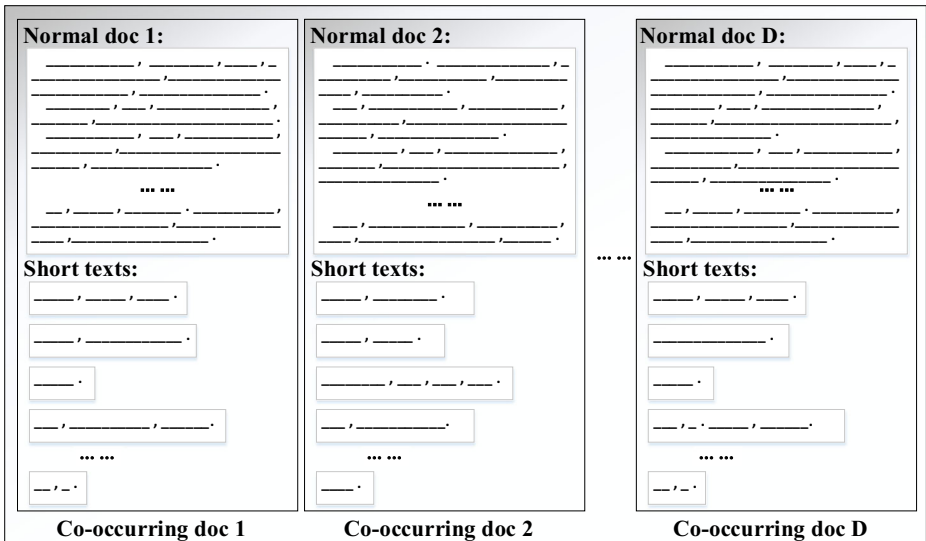


Figure 1 Hierarchical structure of normal documents and short texts

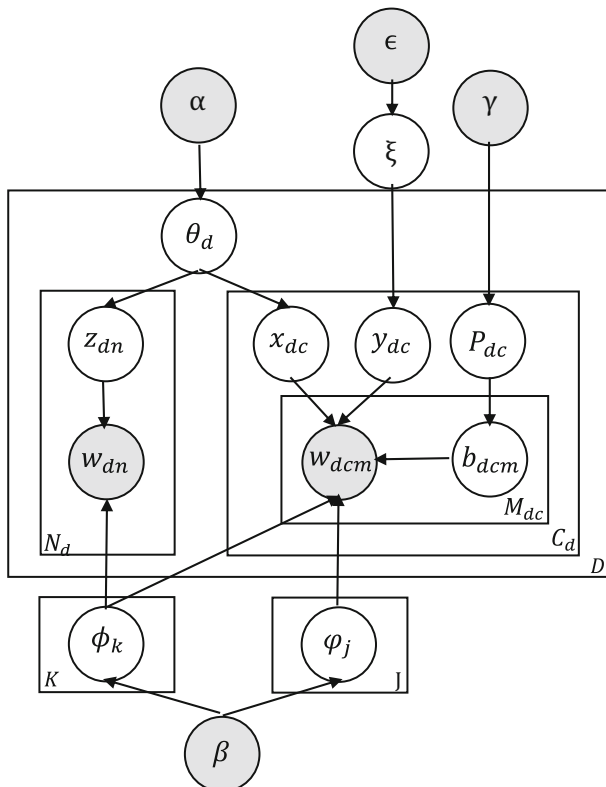
of topic probabilities  $\theta_d = \{\theta_{d1}, \theta_{d2}, \dots, \theta_{dK}\}$ . Each formal topic  $k$  has its own vector of word probabilities  $\phi_k = \{\phi_{k1}, \phi_{k2}, \dots, \phi_{kV}\}$  over a dictionary with size  $V$ , which consists of all distinct words in normal documents and short texts.

Contents in short texts may discuss topics from their corresponding normal documents, and may also discuss some additional issues, such as adding personal opinions. Hence the  $K$  formal topics are insufficient to cover all subjects discussed in the short text corpus. We assume there are another set of  $J$  informal topics, which appear only in short texts, and each informal topic  $j$  has its own vector of word probabilities  $\psi_j = \{\psi_{j1}, \psi_{j2}, \dots, \psi_{jV}\}$ . For the  $c$ th short text following normal document  $d$ , it has a probability distribution  $(p_{dc}, 1 - p_{dc})^\top$  over two topics, a formal topic  $x_{dc}$  and an informal topic  $y_{dc}$ . Here  $p_{dc} \in [0, 1]$  depicts the association probability between the short text and the corresponding normal document, with higher values indicating more consistent relationships.

The graphical representation of normal documents and short texts is illustrated in Figure 2, and the generative process is described below.

For each normal document  $d \in \{1, 2, \dots, D\}$ :

1. Generate topic probabilities  $\theta_d$  from a homogeneous Dirichlet distribution with parameter  $\alpha$ :  $\theta_d \sim Dir(\alpha)$ ;
2. For the  $n$ th word in normal document  $d$ ,  $n \in \{1, 2, \dots, N_d\}$ :
  - (a) Choose a topic  $z_{dn}$  from the  $K$  formal topics with probabilities given by  $\theta_d$ :  $z_{dn} \sim Multi(\theta_d)$ ;



**Figure 2** Graphical representation of COTM

- (b) Choose a word  $w_{dn}$  from the dictionary with probabilities given by  $\phi_{z_{dn}}: w_{dn} \sim Multi(\phi_{z_{dn}})$ .

Then for the  $c$ th short text associated with normal document  $d$ ,  $c \in \{1, 2, \dots, C_d\}$ :

1. Choose the association probability  $p_{dc}$  from a beta distribution with parameter  $\gamma$ :  $p_{dc} \sim Beta(\gamma, \gamma)$ ;
2. Choose a topic  $x_{dc}$  from  $K$  formal topics with probabilities given by  $\theta_d: x_{dc} \sim Multi(\theta_d)$ ;
3. Choose a topic  $y_{dc}$  from  $J$  informal topics with probabilities given by  $\xi$ :  $y_{dc} \sim Multi(\xi)$ ;
4. For the  $m$ th word in the short text,  $m \in \{1, 2, \dots, M_{dc}\}$ :
  - (a) Generate a topic indicator  $b_{dcm}$  with probability given by  $p_{dc}: b_{dcm} \sim Bernoulli(p_{dc})$ ;
  - (b) If  $b_{dcm} = 1$ , the word is chosen with probabilities under the formal topic:  $w_{dcm} \sim Multi(\phi_{x_{dc}})$ .
  - (c) If  $b_{dcm} = 0$ , the word is chosen with probabilities under the informal topic:  $w_{dcm} \sim Multi(\psi_{y_{dc}})$ ;

To complete the specification, we assign homogeneous Dirichlet hyperpriors for  $\phi_k, \psi_j$  and  $\xi$ , i.e.:  $\phi_k \sim Dir(\beta), \psi_j \sim Dir(\beta), \xi \sim Dir(\epsilon)$ .

Here we make a few comments about the model specification. Firstly, unlike normal documents, each of which has its own topic probabilities  $\theta_d$  over the formal topics, we assume all short texts share the same topic probabilities  $\xi$  over the informal topics. This assumption makes the model simpler and thus easier to converge than assuming different topic probabilities for each short text. Secondly, noting that short texts are very concise, we assume each short text only represents two topics, a formal one and an informal one. This assumption also helps to simplify our model setting. Lastly, by using  $p_{dc}$  to depict the topical relationships between short texts and normal documents, we obtain an unsupervised way to detect “spams”, i.e. the short texts whose  $p_{dc}$  are smaller than a predefined threshold can be identified as “spams”.

To our best knowledge, models utilizing the co-occurring relationships between normal documents and short texts to enhance topic learning are still rarely seen in the literature, except for the MSTM and EXTM models proposed by Ma et al. [16]. While both COTM and the models in [16] employ the co-occurring structure in text corpus, there still exists many differences between these two methods:

- Firstly, both MSTM and EXTM allow each short text to have only one topic, which is either derived from the topic distribution of its associated normal document or is one of the extended topics formed by all short texts. However, this assumption for short texts is still rigid because these two circumstances may coexist on one short text. Following a normal document discussing various topics, the corresponding short texts often concentrate on one specific topic (the formal one) and add additional personal opinions (the informal one). Thus, in COTM, we assume each short text is composed of two topics, a formal one derived from the normal documents and the informal one formed only by short texts. This assumption is much closer to the generative process of short texts in real world. In addition, a probability distribution is assumed over the two topics, indicating the correlation relationship between the short text and its associated normal document. Therefore, the topics of short texts are under different level of topical consistence with normal documents, from strongly correlated, partially correlated to

completely irrelevant. In this respect, COTM can be seen as a more general extension of EXTM.

- Secondly, in MSTM and EXTM, the same topic meaning is represented by two set of topics, the master topics, which use vocabulary formed by normal documents, and the slave topics, which use vocabulary formed by short texts. On the contrary, both formal and informal topics have word distributions over the whole vocabulary, which includes all unique words in normal documents and short texts. This more concise assumption for vocabulary can not only significantly reduce the parameter space and computational complexity under a big text corpus, but also integrate the generation of topics in an unified framework and make the artificial summarization of topic meanings easier. Moreover, under this assumption, the topic learning of normal documents can be enhanced by the inclusion of words from the corresponding short texts.

### 2.2 Model inference

In this section, we introduce the Gibbs sampling algorithm for COTM. For normal documents, let  $\mathbf{z}_d = (z_{d1}, z_{d2}, \dots, z_{dN_d})^\top$  and  $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_D\}$ . For all the  $C_d$  short texts associated with normal document  $d$ , let  $\mathbf{b}_{dc} = (b_{dc1}, b_{dc2}, \dots, b_{dcM_{dc}})^\top$ ,  $\mathbf{b}_d = \{\mathbf{b}_{d1}, \mathbf{b}_{d2}, \dots, \mathbf{b}_{dC_d}\}$ ,  $\mathbf{P}_d = \{p_{d1}, p_{d2}, \dots, p_{dC_d}\}$ ,  $\mathbf{x}_d = (x_{d1}, x_{d2}, \dots, x_{dC_d})^\top$  and  $\mathbf{y}_d = (y_{d1}, y_{d2}, \dots, y_{dC_d})^\top$ . Then we have  $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_D\}$ ,  $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_D\}$ ,  $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D\}$ , and  $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_D\}$  for all short texts. Moreover, let  $\Theta = \{\theta_1, \theta_2, \dots, \theta_D\}$ ,  $\Phi = \{\phi_1, \phi_2, \dots, \phi_K\}$  and  $\Psi = \{\psi_1, \psi_2, \dots, \psi_K\}$ . Let  $\mathbf{w}$  represent all words in normal documents and short texts. Given  $\mathbf{w}$  and all the hyperparameters, we can derive the full posterior distribution according to the generative process of COTM:

$$\begin{aligned}
 & f(\mathbf{z}, \mathbf{b}, \mathbf{P}, \mathbf{x}, \mathbf{y}, \Theta, \Phi, \Psi, \xi \mid \mathbf{w}, \alpha, \beta, \gamma, \epsilon) \\
 & \propto \left\{ \prod_{d=1}^D \prod_{k=1}^K \theta_{dk}^{\alpha-1} \right\} \left\{ \prod_{k=1}^K \prod_{v=1}^V \phi_{kv}^{\beta-1} \right\} \left\{ \prod_{j=1}^J \xi_j^{\epsilon-1} \right\} \left\{ \prod_{j=1}^J \prod_{v=1}^V \psi_{jv}^{\beta-1} \right\} \\
 & \left\{ \prod_{d=1}^D \prod_{c=1}^{C_d} p_{dc}^{\gamma-1} (1 - p_{dc})^{\gamma-1} \right\} \left\{ \prod_{d=1}^D \prod_{n=1}^{N_d} \theta_{d,z_{dn}} \phi_{z_{dn},w_{dn}} \right\} \\
 & \left\{ \prod_{d=1}^D \prod_{c=1}^{C_d} \theta_{d,x_{dc}} \xi_{y_{dc}} \right\} \left\{ \prod_{d=1}^D \prod_{c=1}^{C_d} \prod_{m=1}^{M_{dc}} (p_{dc} \phi_{x_{dc},w_{dc}})^{b_{dc}} \right\} \\
 & \left\{ \prod_{d=1}^D \prod_{c=1}^{C_d} \prod_{m=1}^{M_{dc}} [(1 - p_{dc}) \psi_{y_{dc},w_{dc}}]^{1-b_{dc}} \right\}.
 \end{aligned} \tag{1}$$

Given the full posterior distribution in (1), we can easily get the full conditional posterior distributions for  $\Theta, \Phi, \Psi, \xi$  and  $\mathbf{P}$ , which are all Dirichlet and conjugate with their priors. Therefore, we develop a collapsed Gibbs sampling algorithm by integrating out these parameters from the posterior distribution, and only need to update  $\mathbf{z}, \mathbf{x}, \mathbf{y}$  and  $\mathbf{b}$  in each iteration. Details of deriving the collapsed Gibbs sampling algorithm can be seen in [Appendix](#).

For the  $n$ th word in normal document  $d$ , the full conditional distribution of  $z_{dn}$  in the collapsed Gibbs sampling algorithm is:

$$f(z_{dn} = k \mid \cdot) \propto \left( l_{dk;-dn}^{(1)} + g_{dk}^{(1)} + \alpha \right) \frac{l_{k,w_{dn};-dn}^{(2)} + g_{k,w_{dn}}^{(2)} + \beta}{l_{k;-dn}^{(2)} + g_k^{(2)} + V\beta}, \tag{2}$$

where the subscript “ $-dn$ ” indicates counts excluding the  $n$ th word in normal document  $d$ ,  $l_{dk}^{(1)}$  and  $g_{dk}^{(1)}$  denote the number of words in normal document  $d$  or the number of short texts

following normal document  $d$  that are associated with formal topic  $k$ ,  $l_{kv}^{(2)}$  and  $g_{kv}^{(2)}$  denote the number of times word  $v$  is associated with formal topic  $k$  in all normal documents and short texts,  $l_{k\cdot}^{(2)}$  and  $g_{k\cdot}^{(2)}$  are the sum of  $l_{kv}^{(2)}$  and  $g_{kv}^{(2)}$  over all words  $v$  in the vocabulary.

For  $x_{dc}$  and  $y_{dc}$  in the  $c$ th short text associated with normal document  $d$ , their full conditional distributions in the collapsed Gibbs sampling algorithm are:

$$f(x_{dc} = k \mid \cdot) \propto \left( l_{dk}^{(1)} + g_{dk;-dc} + \alpha \right) \frac{\prod_{v \in \Lambda_{dc}} \prod_{m=1}^{q_{dcv}^{(1)}} \left( l_{kv}^{(2)} + g_{kv;-dc} + m - 1 + \beta \right)}{\prod_{m=1}^{s_{dc}^{(1)}} \left( l_{k\cdot}^{(2)} + g_{k\cdot;-dc} + m - 1 + V\beta \right)}, \tag{3}$$

$$f(y_{dc} = j \mid \cdot) \propto \left( h_{j;-dc} + \epsilon \right) \frac{\prod_{v \in \Lambda_{dc}} \prod_{m=1}^{q_{dcv}^{(2)}} \left( g_{jv;-dc} + m - 1 + \beta \right)}{\prod_{m=1}^{s_{dc}^{(2)}} \left( g_{j\cdot;-dc} + m - 1 + V\beta \right)}, \tag{4}$$

where the subscript “ $-dc$ ” indicates counts excluding the  $c$ th short text following normal document  $d$ ,  $\Lambda_{dc}$  is the set of unique words appearing in the  $c$ th short text following normal document  $d$ ,  $q_{dcv}^{(1)}$  and  $q_{dcv}^{(2)}$  denote the the number of times word  $v$  appears in the  $c$ th short text following normal document  $d$  and are associated with formal topics or informal topics respectively,  $h_j$  denotes the number of short texts that are associated with informal topic  $j$ ,  $g_{jv}^{(3)}$  denotes the number of times word  $v$  is associated with informal topic  $j$  in all short texts,  $s_{dc}^{(1)}$  and  $s_{dc}^{(2)}$  are summation of  $q_{dcv}^{(1)}$  and  $q_{dcv}^{(2)}$  over all unique words in  $\Lambda_{dc}$ .

For the  $m$ th word in the  $c$ th short text following normal document  $d$ , the full conditional distribution of  $b_{dcm}$  in the collapsed Gibbs sampling algorithm is:

$$f(b_{dcm} = 1 \mid \cdot) \propto \frac{l_{x_{dc},w_{dcm}}^{(2)} + g_{x_{dc},w_{dcm};-dcm} + \beta}{l_{x_{dc}\cdot}^{(2)} + g_{x_{dc}\cdot;-dcm} + V\beta} \left( s_{dc;-dcm}^{(1)} + \gamma \right), \tag{5}$$

$$f(b_{dcm} = 0 \mid \cdot) \propto \frac{g_{y_{dc},w_{dcm};-dcm} + \beta}{g_{y_{dc}\cdot;-dcm}^{(3)} + V\beta} \left( s_{dc;-dcm}^{(2)} + \gamma \right), \tag{6}$$

where the subscript “ $-dcm$ ” indicates counts excluding the  $m$ th word in the  $c$ th short text following normal document  $d$ ,  $g_{y_{dc}\cdot}^{(3)}$  is the sum of  $g_{y_{dc},v}^{(3)}$  over all words in the dictionary. Equations (5) and (6) are then normalized to sum up to one to get the full conditional posterior probabilities for  $b_{dcm} = 1$  and  $b_{dcm} = 0$ .

We can compute  $\Theta$ ,  $\Phi$ ,  $\Psi$  and  $P$  using the first posterior draw of  $z$ ,  $x$ ,  $y$  and  $b$  after convergence of the collapsed Gibbs sampling algorithm.

$$\hat{\theta}_{dk} = \frac{l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha}{l_{d\cdot}^{(1)} + g_{d\cdot}^{(1)} + K\alpha}, \tag{7}$$

$$\hat{\phi}_{kv} = \frac{l_{kv}^{(2)} + g_{kv}^{(2)} + \beta}{l_{k\cdot}^{(2)} + g_{k\cdot}^{(2)} + V\beta}, \tag{8}$$

$$\hat{\psi}_{jv} = \frac{g_{jv}^{(3)} + \beta}{g_{j\cdot}^{(3)} + V\beta}, \tag{9}$$

$$\hat{p}_{dc} = \frac{s_{dc}^{(1)} + \gamma}{s_{dc}^{(1)} + s_{dc}^{(2)} + 2\gamma}. \tag{10}$$



**Table 1** Time complexity and the number of in-memory variables in LDA and COTM

	Time complexity	The number of in-memory variables
LDA	$O(N_{iter}D(K + J)(\bar{N} + \bar{C}\bar{M}))$	$D(K + J)(1 + \bar{C}) + V(K + J) + D(\bar{N} + \bar{C}\bar{M})$
COTM	$O(N_{iter}D(K\bar{N} + \bar{C}(K + J + 2\bar{M})))$	$DK + V(K + J) + D(\bar{N} + \bar{C}(4 + \bar{M}))$

### 2.3 Model complexity

To illustrate the computational complexity of COTM, we show its running time and memory requirements and make comparison with the basic LDA model. We denote by  $\bar{C}$  the average number of short texts following each normal document,  $\bar{N}$  the average length (number of words) of normal documents,  $\bar{M}$  the average length of short texts, and  $N_{iter}$  the number of iterations in Gibbs sampling. For simplicity of calculations, we further assume each normal document has the same number of short texts  $\bar{C}$ , each normal document has the same length  $\bar{N}$ , and each short text has the same length  $\bar{M}$ . To ensure fair comparison with the same set of texts and the same number of topics, we compare COTM with the LDA model trained by Gibbs sampling<sup>1</sup> to both normal documents and short texts with  $K + J$  topics. The time complexity and number of in-memory variables in the Gibbs sampling procedure of the two models are listed in Table 1.

For LDA, the assignment of each topic requires computational time in the order  $O(K + J)$ . LDA draws a topic for each word in the text corpus, with an overall time complexity  $O(N_{iter}D(K + J)(\bar{N} + \bar{C}\bar{M}))$ . For COTM, there are three sampling steps. In the first step, COTM draws a topic for each word in normal documents, which requires computational time in the order  $O(N_{iter}DK\bar{N})$ . The second step involves drawing a formal topic and an informal topic for each short text, which requires computational time in the order  $O(N_{iter}D\bar{C}(K + J))$ . In the final step, COTM draws the binary topic indicator  $b_{dcm}$  for each word in the short texts, which requires computational time in the order  $O(N_{iter}D\bar{C}2\bar{M})$ . Therefore, the overall time complexity of COTM is  $O(N_{iter}D(K\bar{N} + \bar{C}(K + J + 2\bar{M})))$ . The difference in the order of computational complexity between LDA and COTM is  $O(N_{iter}(DJ\bar{N} + D\bar{C}\{(K + J)\bar{M} - (K + J) - 2\bar{M}\}))$ . Noting  $K + J + 2\bar{M} \ll (K + J)\bar{M}$ , the time complexity of COTM has smaller order than that of LDA.

In the two models, count matrices and topic assignments need to be kept in memory. In LDA, the variables that need to be stored are: the count matrix for the number of words in each normal document or short text that are associated with each topic, the count matrix for the number of times that each word in the dictionary is associated with each topic, and the topic assignment for each word in the corpus. Hence the overall required memory size is  $D(K + J)(1 + \bar{C}) + V(K + J) + D(\bar{N} + \bar{C}\bar{M})$ . In COTM, the count matrices  $l_{dk}^{(1)} + g_{dk}^{(1)}$ ,  $l_{kv}^{(2)} + g_{kv}^{(2)}$ ,  $g_{jv}^{(3)}$ ,  $s_{dc}^{(1)}$  and  $s_{dc}^{(2)}$  need to be stored, taking up memory size  $DK + V(K + J) + 2D\bar{C}$ . Moreover, the topic assignments  $z$ ,  $x$ ,  $y$  and  $b$  need to be stored, taking up memory size  $D\bar{N} + 2D\bar{C} + D\bar{C}\bar{M}$ . Hence, the overall required memory size for COTM is  $DK + V(K + J) + D(\bar{N} + \bar{C}(4 + \bar{M}))$ . The difference in required memory size between LDA and COTM is  $D(J + (K + J) - 4)\bar{C}$ , which is usually very large. Thus the required memory size for COTM is less than that for LDA.

<sup>1</sup><http://gibbslda.sourceforge.net/>

### 3 Online algorithm for COTM

In real-world applications, normal documents and their co-occurring short texts are constantly generated, which requires topic modeling algorithms to deal with large volume of data streams. Batch algorithms have high computational and memory cost, and are not efficient. As a result, we introduce an online algorithm for COTM, referred to as oCOTM, to deal with the online-learning task. Compared with batch COTM, the online algorithm only needs to store a small amount of data, and topics can be constantly updated over data streams.

The oCOTM algorithm is inspired by the online LDA algorithm proposed in [1]. It assumes documents are divided into successive time slices, e.g., each time slice being an hour or a day. The general idea of oCOTM is to fit a COTM model with  $K$  formal topics and  $J$  informal topics on normal documents and short texts at each time slice, and the counts of words in topics (i.e.  $l_{kv}^{(2)}, g_{kv}^{(2)}, g_{jv}^{(3)}$ ) at the current time slice would be used to update parameters in priors for topics' word probabilities in COTM at the next time slice.

Let  $V^{(t)}$  denote the size of dictionary at time slice  $t$ , where the dictionary expands that at time slice  $t - 1$  by the new words appearing in time slice  $t$ . Let  $\beta_{k;\text{for}}^{(t)}$  and  $\beta_{j;\text{inf}}^{(t)}$  respectively denote  $V^{(t)}$ -dimensional vectors used in Dirichlet priors for formal topic  $k$  and informal topic  $j$  at time slice  $t$ , where the components corresponding to the new words equal  $\beta$ . The collapsed Gibbs sampling algorithm in Section 2 is carried out, with  $\beta$  replaced by  $\beta_{kv;\text{for}}^{(t)}$  or  $\beta_{jv;\text{inf}}^{(t)}$ , and the count matrices  $l_{dk}^{(1)}, g_{dk}^{(1)}, l_{kv}^{(2)}, g_{kv}^{(2)}, g_{jv}^{(3)}, s_{dc}^{(1)}$  and  $s_{dc}^{(2)}$  calculated using only normal documents and short texts at time slice  $t$ .

After convergence of the collapsed Gibbs sampling algorithm, the first posterior draw of  $z, \mathbf{x}, \mathbf{y}$  and  $\mathbf{b}$  is used to calculate the word counts  $l_{kv}^{(2)(t)}, g_{kv}^{(2)(t)}$  and  $g_{jv}^{(3)(t)}$ . These counts are used to adjust the prior vectors for the next time slice by setting:

$$\beta_{kv;\text{for}}^{(t+1)} = \beta_{kv;\text{for}}^{(t)} + \lambda \left( l_{kv}^{(2)(t)} + g_{kv}^{(2)(t)} \right), \tag{11}$$

$$\beta_{jv;\text{inf}}^{(t+1)} = \beta_{jv;\text{inf}}^{(t)} + \lambda g_{jv}^{(3)(t)}, \tag{12}$$

where  $\lambda \in [0, 1]$  is a decay parameter, indicating the strength of influence of historical topic information. When  $\lambda = 1$ , we simply accumulate the historical counts of topic assignments without any decay; when  $\lambda = 0$ , the COTM models trained at different time slices are independent. At the initial time slice, we set all entries in  $\beta_{k;\text{for}}^{(t)}$  and  $\beta_{j;\text{inf}}^{(t)}$  as a constant  $\beta$ . Then these prior vectors would be updated at the end of each time slice and the historical information would be involved in model fitting at later time.

We now compare the complexities of batch and online algorithms of COTM. Assume there are  $D^{(t)}$  normal documents in time slice  $t$ . Let  $D^{(1:t)} = D^{(1)} + \dots + D^{(t)}$  denote the cumulative number of normal documents up to time  $t$ . For simplicity, we further assume all short texts corresponding to a normal document in time slice  $t$  are published in time slice  $t$ . The batch COTM algorithm needs to take account of all normal documents and short texts up to time  $t$ . It would require computational time in the order  $O(N_{iter} D^{(1:t)} (K\bar{N} + \bar{C}(K + J + 2\bar{M})))$ , and memory size  $D^{(1:t)} K + V^{(t)} (K + J) + D^{(1:t)} (\bar{N} + \bar{C}(4 + \bar{M}))$ . The online COTM algorithm needs only to take account of normal documents and short texts in time slice  $t$ . It would require computational time in the order  $O(N_{iter} D^{(t)} (K\bar{N} + \bar{C}(K + J + 2\bar{M})))$ , and memory size  $D^{(t)} K + V^{(t)} (K + J) + D^{(t)} (\bar{N} + \bar{C}(4 + \bar{M}))$ . The computational complexity and memory consumption are compared in Table 2.

**Table 2** Time complexity and the number of in-memory variables of batch and online COTM algorithms in time slice  $t$ 

	Time complexity	The number of in-memory variables
batch COTM	$O(N_{iter} D^{(1:t)}(K\bar{N} + \bar{C}(K + J + 2\bar{M})))$	$D^{(1:t)}K + V^{(t)}(K + J) + D^{(1:t)}(\bar{N} + \bar{C}(4 + \bar{M}))$
oCOTM	$O(N_{iter} D^{(t)}(K\bar{N} + \bar{C}(K + J + 2\bar{M})))$	$D^{(t)}K + V^{(t)}(K + J) + D^{(t)}(\bar{N} + \bar{C}(4 + \bar{M}))$

## 4 Experiments

### 4.1 Experimental settings

**Datasets.** The effectiveness of our approach is evaluated over two text datasets with co-occurring structure.

- **NetEase** collection includes news articles and reader comments crawled from the most popular Chinese news publishing platform.<sup>2</sup> All the texts crawled are published between May 1st, 2015 and May 1st, 2016.
- **Sina** collection includes blog posts and user comments crawled from a famous Chinese blog platform.<sup>3</sup> All the texts crawled are published between Jan 1st, 2016 and May 1st, 2016. Each blog post is assigned to one of eight categories by its author, as illustrated in Figure 3a.

All the datasets have been made to be public.<sup>4</sup> The raw texts are mainly written in Chinese and we take the following preprocessing procedure to obtain clean text corpus. Firstly, we erase non-Chinese characters, punctuations and convert traditional Chinese characters to simplified Chinese characters. Secondly, we segment sentences into word sequences using an open source package NLPIR.<sup>5</sup> Finally, we remove stop words, low frequency words and normal documents followed by no short texts. After preprocessing, the basic statistics of the two datasets are listed in Table 3, including the numbers of normal documents and short comments, the average lengths of normal documents and short comments, and the number of unique Chinese words. Figure 3b also illustrates the distribution of counts of short comments (in logarithm) following normal documents for NetEase data. The distribution follows power-law and has a heavy tail, indicating that while some news articles gain great popularity among news readers, most of them are only followed by a few short comments.

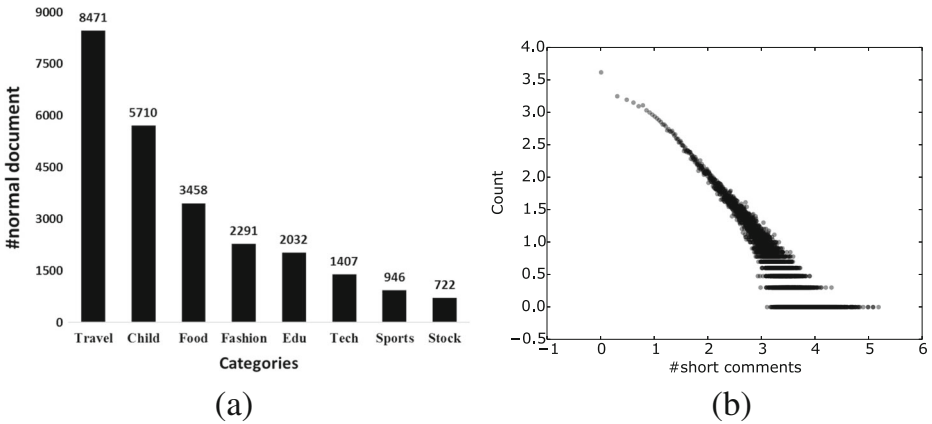
We first evaluate the batch COTM algorithm on a randomly sampled dataset of 10% normal documents and their corresponding comments in NetEase and Sina collections, since both of the datasets are too large to be processed efficiently by batch algorithms. Then the online algorithm of COTM is evaluated over the whole datasets of NetEase and Sina collections. For the online algorithm, the time periods in the data sets are equally divided

<sup>2</sup><http://news.163.com/>

<sup>3</sup><http://blog.sina.com.cn/>

<sup>4</sup><https://pan.baidu.com/s/1boVox3p>

<sup>5</sup><https://pypi.python.org/pypi/PyNLPIR/>



**Figure 3** **a** Categories of Sina blog posts and **b** Distribution of counts of comments for NetEase news articles

into  $T = 20$  time slices, with each time slice roughly equal to 18 days for the NetEase data or 6 days for the Sina data.

**Baseline methods** In COTM, the semantic meanings of normal documents is covered by the formal topics, and the semantic meanings of short texts is covered by both formal topics and informal topics. We compare topics learned by COTM with the following state-of-art baselines.

- **LDA-P**: the standard LDA model trained by Gibbs sampling and applied to pseudo-documents obtained by aggregating each normal document with its corresponding short texts.
- **LCTM-P**: the LCTM model trained by Gibbs sampling and applied to word2vec representation [19] of the pseudo-documents obtained by aggregating each normal document with its corresponding short texts.
- **BTM-B**: the standard BTM model trained by Gibbs sampling and applied to the corpus including both normal documents and short texts and treating them equally.
- **PTM-B**: the standard PTM model trained by Gibbs sampling and applied to the corpus including both normal documents and short texts and treating them equally.
- **EXTM**: the EXTM model trained by Gibbs sampling and applied to the corpus including both normal documents and short texts.

The online algorithm of COTM is also compared with online implementations of LDA [1] and BTM [5], since there are no online versions of the other alternatives:

**Table 3** Basic statistics of NetEase dataset and Sina dataset

Dataset	NetEase	Sina
# of docs ( $D$ )	88,420	25,037
Avg. doc len ( $\bar{N}$ )	359.03	569.65
# of comments	53,555,834	973,120
Avg. comm len ( $\bar{M}$ )	6.73	18.58
# of words ( $V$ )	154,729	118,373

- **oLDA-B**: the online algorithm of LDA applied to the corpus including both normal documents and short texts and treating them equally. Here the algorithm uses the counts of words in topics at the current time slice to update parameters in priors for topics' word probabilities ( $\beta$ ) for the next time slice.
- **oLDA-S**: the online algorithm of LDA applied to short texts.
- **oBTM-S**: the online algorithm of BTM<sup>6</sup> applied to short texts. Here the algorithm fits a BTM model in each time slice, and uses the counts of topics in the corpus and the counts of words in topics at the current time slice to update parameters in priors for the corpus' topic probabilities ( $\alpha$ ) and topics' word probabilities ( $\beta$ ) for the next time slice.
- **iBTM-S**: the incremental algorithm of BTM applied to short texts. Here the algorithm updates prior parameters continuously whenever a piece of text arrives.

For the online LDA algorithm, we do not consider aggregating normal documents and their corresponding short texts into pseudo-documents, because a normal document and its corresponding short texts may not be published in the same time slice. We run online algorithms of BTM only on short texts, because BTM suffers from expensive computational cost and memory explosion when applied to normal documents.

To make fair comparisons, all the methods are implemented in C++, including the batch COTM algorithm<sup>7</sup> and the online COTM algorithm.<sup>8</sup>

The hyperparameters for all baseline models and the online implementations are set to default values. For COTM, results obtained in various hyperparameter settings show little difference, and we set  $\alpha = 0.5$ ,  $\beta = 0.1$ ,  $\gamma = 0.5$  and  $\epsilon = 0.5$  for illustration. In all the methods, Gibbs sampling is run for 1000 iterations, which is enough for convergence. The decay weight  $\lambda$  for online methods are all set to be 1.

**Measurements** Model performance is evaluated in two perspectives: the quality of learned topics, and the quality of topic representation of documents.

We use coherence score [20] to measure the quality of topics learned by each method. Given any topic and its top  $L$  words  $V = (v_1, v_2, \dots, v_L)^T$  ordered by  $\phi_k$  or  $\psi_j$ , the coherence score is defined as:

$$CS(V) = \sum_{l=2}^L \sum_{l'=1}^{l-1} \log \frac{F(v_l, v_{l'}) + 1}{F(v_{l'})}, \quad (13)$$

where  $F(v)$  is the number of relevant documents including word  $v$ ,  $F(v, v')$  is the number of relevant documents including both words  $v$  and  $v'$ . The general idea of this metric is that we believe words belonging to the same topic tend to co-occur within the same document. Therefore topics with higher coherence scores imply better developed methods. Note that this definition is consistent with the basic assumption of BTM, i.e., words co-occurring more frequently should be more possible to belong to the same topic, thus BTM has inherent advantage under this evaluation metric [5].

To evaluate the quality of topic representation of documents, we investigate how much the documents' topic probabilities can help discriminate documents in different clusters or classes. For LDA and LCTM, document  $d$ 's topic proportions  $\theta_d$  are used as features. For PTM, topic proportions of each document are those of its associated pseudo

<sup>6</sup><https://github.com/xiaohuiyan/OnlineBTM>

<sup>7</sup><https://github.com/dongxiexidian/hdLDA>

<sup>8</sup><https://github.com/dongxiexidian/ohdLDA>

document. In BTM, the topic proportions of each document are derived using the topic indicators  $z$  [5, 27]. However, [14, 25] have validated that topic proportions of documents obtained by using post inference method are critical for downstream applications. In this context, we do not compare BTM models in document clustering and classification. In COTM, the formal topic proportions  $\theta_d$  are used as features in clustering and classifying normal documents. For each short text, a  $(K + J)$ -dim vector of pseudo topic proportions  $\tilde{\theta}_{dc}$  is created by setting the entry corresponding to the formal topic  $x_{dc}$  to  $p_{dc}$ , and the entry corresponding to the informal topic  $y_{dc}$  to  $1 - p_{dc}$ , and these topic proportions are then used as features in clustering and classifying short texts. Similar with COTM, in EXTM, we use the proportions of master topics to classify normal documents and also transfer the specific topic of each short text into a pseudo vector to classify short texts.

In document clustering, K-means algorithm is performed under different number of clusters, and the pseudo F index [4], which describes the ratio of between-cluster variance to within cluster variance, is used to evaluate the performance of clustering.

The pseudo  $F$  index is defined as follows. Let  $\|\cdot\|$  denote the Euclidean distance, let  $\Omega_g$  denote the set of indices of documents in the  $g$ th cluster, and let  $|\Omega|$  denote the number of normal documents in cluster  $\Omega$ . For now, let  $\theta_d$  denote the general topic proportions for documents  $d$ , including the case of pseudo topic proportions. For the  $g$ th cluster, we denote  $\bar{\theta}_g = \frac{1}{|\Omega_g|} \sum_{d \in \Omega_g} \theta_d$  as the average topic proportions of documents in this cluster; we denote  $\bar{\theta} = \frac{1}{D} \sum_{d=1}^D \theta_d$  as the average topic proportions of all documents. Then the within-group sum of squares  $SSW$  and between-group sum of squares  $SSG$  can be derived as:

$$SSW = \sum_{g=1}^G \sum_{d \in \Omega_g} \|\theta_d - \bar{\theta}_g\|^2, \tag{14}$$

$$SSG = \sum_{d=1}^D \|\theta_d - \bar{\theta}\|^2 - SSW, \tag{15}$$

where  $G$  is the number of clusters. Then the pseudo  $F$  index is calculated as

$$pseudo\ F = \frac{SSG/(G - 1)}{SSW/(D - G)}. \tag{16}$$

Larger values of pseudo  $F$  index indicate that the clusters are better separated, implying that the topic representations of documents is of higher quality.

In document classification, we use the SVM classifier LIBLINEAR[8] with 10-fold cross validation. Methods resulting in better classification accuracy indicate better topic representations of documents.

## 4.2 Evaluation of batch COTM

### 4.2.1 Evaluating formal topics

We first evaluate the quality of learned formal topics, and then perform clustering and classification of normal documents to evaluate the quality of document representation using formal topic proportions. Since BTM and PTM are originally proposed to deal with short texts, here we only compare COTM with models which are designed for modeling normal documents, such as LDA-P, LCTM-P and EXTM.

**Comparison of words under topics** In COTM, formal topics are mixtures of words from both normal documents and short texts. While in LCTM-P, topics are mixtures of concepts rather than words, we only compare the formal topics learned by COTM with those learned by LDA-P and EXTM. In the experiment, the numbers of formal (master) and informal

(extended) topics for COTM (EXTM) are set to be  $K_{COTM} = 100$  and  $J_{COTM} = 50$ . To make a fair comparison, we set the number of topics for LDA-P to be 150, since it is applied to pseudo-documents that include both normal documents and their corresponding short texts.

We randomly select three topics shared by the three methods to make the comparison. We follow [3] to proceed this selection. We first create for each method a topic word set including the top five words with highest probabilities under each topic. We then get the intersection set of the three topic word sets. Finally, we randomly select three words from the intersection set. For each selected word, we use the topics whose top five words include the given word as illustrative examples.

For the Netease data, the three selected words are “airplane”, “cellphone” and “student”. Table 4 shows topics selected by the word “airplane” for the three methods. In the first row which lists the top twenty words with highest probabilities under the selected topics, we find “airplane”, “aviation” and “airport” are among the top words in all three methods. This indicates that all three topics discuss aviation. However, the top word set of LDA-P includes words “legitimate” and “France”, which have little to do with aviation. As for EXTM, its master topic has more irrelevant words, such as “Jackie-Chen”, “France” and “Germany”, which are names of super stars or countries. Since EXTM uses separate vocabularies for normal documents and short texts, master topics are only represented by words appearing in normal documents and less influenced by short texts. Results for COTM are better than those for LDA-P and EXTM, since most of the top words for COTM are closely related to aviation. Moreover, the formal topic in COTM is enhanced by including words “China”, “design” and “aero-engine” from user comments, as the underdeveloped manufacturing of “aero-engine” in Chinese aircraft industry has long been a hot issue discussed among Chinese netizens.

Table 5 shows topics selected by the word “cellphone” for the three methods. The top twenty words listed in the first row include words “cellphone”, “Apple” and “Mi” (a Chinese mobile internet company), which indicates that all three topics discuss mobile industry.

**Table 4** Topics selected by the word “airplane” in NetEase collection

LDA-P	EXTM	COTM
<b>airplane</b> airport aviation voyage pilot airlines fly helicopter passenger safety flight airliner legitimate sorties crash G&M fighter take-a-seat France captain  probability approval survivor drop federalism intercontinental celebrate Wuhan black-hawk fortunately start-up uniform wonderful circuses USAF kilometer publish mentality Hollande bleed	<b>airplane</b> aviation airport local passenger voyage G&M France flight airline Germany airliner design Jackie-Chan departure place pilot Shanghai test-flight passenger  Norway city landing small-town sea economic take-off miles heavy cultural-travel check-in now car trips runaway power equipment process usual prohibit	<b>airplane</b> airport drone voyage aviation China design Hefei aero-engine airlines airliner passenger flight research crash G&M route Boeing parachute helicopter  express Air-Malaysia deafening amount nervous DJI young sometimes pull-off traffic-police worried air kilometer piloting meteorological Hongkong life-risk propeller freight-transport district

The first row lists the top 20 words with highest probabilities, while the second row lists non-top words ranked from 501 to 520

**Table 5** Topics selected by the word “cellphone” in NetEase collection

LDA-P	EXTM	COTM
<b>cellphone</b> Apple Mi product user Samsung software system Microsoft computer Huawei company function hardware App watch smart Google equipment usage  patent-fee love dual-cards utilize online-banking rely against wear power-up search-engine landed replace account number give-up use-proxy formal recently official contact	<b>cellphone</b> market brand Apple consumer watch product Mi conference Samsung tradition equipment Huawei screen convention reveal sales Android release mobile  pick-up open kilometers locked inserted promote link known executive ghost compete exams rely Nike sectionary cheat school solely rely famous	<b>cellphone</b> Apple Mi system Samsung Microsoft Green product Huawei computer domestic function software support watch Lenovo Google Android device air-conditioner  electronic GPS clamshell-phone desktop LeTV diversification salute test lifetime amusement smuggled-goods workmanship market-share chairman Japan safety sensor malware telecommunication

The first row lists the top 20 words with highest probabilities, while the second row lists non-top words ranked from 501 to 520

When comparing LDA-P and EXTM, we find that the topic learned by LDA-P is more concentrated on attributes and brands of cellphones. This finding indicates that aggregating user comments with news articles can enhance topic learning in LDA-P, while using separate vocabularies for normal documents and short texts would lead to EXTM borrowing less content information from short texts. The COTM model achieves comparable results with LDA-P, but also includes words such as “Lenovo” (a Chinese technology company) and “Green” (a Chinese electric appliances company). Interestingly, Lenovo is a major brand in the Chinese cellphone market, and Green has recently released two unsuccessful cellphone models.

From the above two comparisons, we find that the formal topics learned by LDA-P and COTM outperform those learned by EXTM. This indicates that when most short texts are topically related to the corresponding normal documents, both LDA-P and COTM are able to enhance the learning of formal topics by using the whole vocabulary that consists of normal documents and short texts. However, it is not the same case when most short texts are topically irrelevant to the normal documents, as demonstrated below.

Table 6 shows topics selected by the word “student” for the three methods. The top twenty words listed in the first row include words “student”, “school” and “teacher”, which indicates that all three topics discuss school life. EXTM performs well with only a few less relevant words, such as “American” and “management”. Results for LDA-P are worse than those for EXTM, with the top words including names of Chinese cities, provinces or companies, like “Kunming”, “Suqian”, “Yunnan” and “Zhonghao”. This can be largely attributed to highly spammed user comments following news articles related to school life. In this case, only using vocabulary formed by normal documents and modeling spam comments by extended topics could lead to more prominent master topics in EXTM. The COTM model not only separates topically relevant contents and spam contents in user comments, but also enhances the learning of formal topics with relevant information from user comments. As a result, COTM achieves the best performance, with its top twenty words mostly related to school life.



**Table 6** Topics selected by the word “student” in NetEase collection

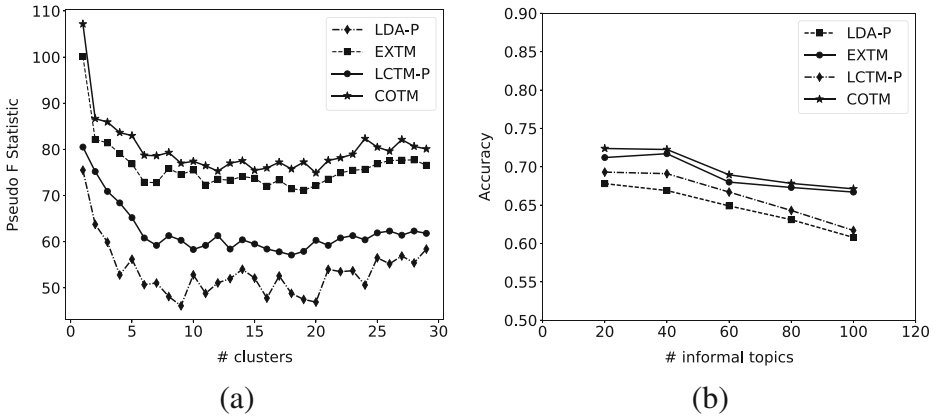
LDA-P	EXTM	COTM
<b>student</b> teacher school hatred	<b>student</b> school teacher computer	<b>student</b> teacher school child
Kunming university education	student head-teacher college	university education headmaster
headmaster child Suqian	management guidelines American	schoolmate elementary-school
Yunnan schoolmate parent	undergraduate campus college	parents teach graduation leader
teacher leader high-school	parents high score games dormitory	undergraduate high-school
Zhonghao undergraduate	classrooms head-teacher	learn college-entrance-examination
graduation elementary-school		exam teacher head-teacher
impression outside-school plan	province teach new wall	dining-hall complementary PhD
matriculate teacher-and-student	everyone midway library	hard-work Oxford family grade
lord kick establish accident	desks classmates run positive	calculate the-whole-school
veteran-cadre business-school	facility equipment cellphone	agricultural-university Michigan
art-department humiliate guilty	habits homework play movie	college-entrance PE energy
rain glory name-card agree	company multi-level	makeup-lesson teaching child
people’s-congress joyful		Junior similarly number-of-people

For a high quality topic, its non-top words should be semantically related to its top words as much as possible. For topics selected by “airplane”, “cellphone” and “student”, the non-top words whose probabilities ranked from 501 to 520 are listed in the second rows of Tables 4, 5 and 6.

In Table 4, the non-top words for COTM are more related to aviation than those for LDA-P and EXTM. In Table 5, the non-top words for COTM reflect broader issues related to information technology, and are more relate to cellphones than the non-top words for LDA-P and EXTM. In Table 6, the superiority of COTM is even more obvious, with most of its non-top words relevant to school life.

**Comparison of clustering and classification performance** To evaluate the quality of topic representation of normal documents, we compare the performance of using topic proportions derived by LDA-P, LCTM-P, EXTM and COTM to cluster normal documents. Similar to LDA-P, we set the number of topics for LCTM-P to be  $K_{LCTM-P} = 150$  since it is also applied to pseudo-documents which aggregates normal documents and their corresponding short texts. Figure 4a shows the pseudo  $F$  values under different number of clusters for all these methods applied to the NetEase data. We find that LDA-P and LCTM-P achieve smaller pseudo  $F$  values than EXTM and COTM, which indicates that the indiscriminative inclusion of short comments by using pseudo-documents has weakened topic representation of normal documents. On the contrary, by separating topically relevant contents from topically irrelevant contents in short texts, COTM and EXTM achieves higher pseudo  $F$  values. Moreover, COTM performs better than EXTM by consistently obtaining prominent formal topics under circumstances that short texts are topically relevant or irrelevant to their corresponding normal documents.

There are class labels for the normal documents in the Sina dataset, so we use document classification performance to further compare the topic representation of normal documents achieved by LDA-P, LCTM-P, EXTM and COTM. In the experiment, the formal (master) topic numbers for COTM (EXTM) are set to be 100, and the number of informal (extended) topics vary from 20 to 100 with a step of 20 topics. For LDA-P and LCTM-P, their topic



**Figure 4** Comparison of COTM with LDA-P, LCTM-P and EXTM algorithms in **a** clustering NetEase news and **b** classifying Sina blog posts

numbers are set to be the summation of formal topics and informal topics in COTM. Under each setting, we use topic proportions of normal documents to classify blog posts into 8 categories. From the results shown in Figure 4b, we find both COTM and EXTM show superiority against LDA-P and LCTM-P, and COTM consistently outperforms EXTM in all experimental settings.

#### 4.2.2 Evaluating informal topics

To evaluate the quality of informal topics learned by COTM, we make comparisons with BTM-B and PTM-B, since they are demonstrated to have good performances in modeling short texts[5, 30]. In the following evaluations, BTM-B, PTM-B and COTM are all applied to the corpus including both news articles and short comments in the NetEase dataset. The number of topics in COTM are  $K_{COTM} = 100$ ,  $J_{COTM} = 50$ , and those for BTM-B and PTM-B are set as 150.

Following the same strategy used above, two words “judgement” and “nation” are selected from the interaction of topic word sets of BTM-B, PTM-B and COTM, and Table 7 shows the top twenty words under the correspondingly selected topics. In the first row of Table 7, the topics selected by word “judgement” are related to the league matches of China Basketball Association (CBA). Comparing the informal topic of COTM with those matched topics of BTM-B and PTM-B, we find that the topic learned by COTM has discovered more technical details of basketball playing. This difference can be attributed to the fact that the informal topics in COTM may have higher probabilities over the words that only appear in short texts. As a result, the informal topics in COTM tend to reflect more flexible meanings, such as more details of the related issues, or expression of personal opinions. These characteristics are further validated in the second row of Table 7: topics discovered by the three methods are all related to international relationships, but the one extracted by COTM talks more about relations across Taiwan strait, which have long been a hot issue discussed among Chinese netizens.

For further validation of the characteristics of informal topics discovered by COTM, Table 8 shows two unique topics only discovered by COTM. The first row represents a topic of rude talking, and the second row represent a topic of mutual judgements, i.e., users making judgements about each other. These two topics can be commonly found in

**Table 7** Topics selected by the words “judgement” and “nation” in the NetEase dataset

BTM-B	PTM-B	COTM
fans Beijing <b>judgement</b> national cheer Guangdong game cup club Liaoning team win speak support win-first-place below champion people hope foreign-aid	Liaoning <b>judgement</b> Beijing match team club champion win goal people playoffs game best prospect cup inspiring manager association first march	foul whistler <b>judgement</b> Marbury Liaoning Hudson Beijing finally defend ball fan penalty-shot attack layup break player final-quarter penalty obvious goal
China USA <b>nation</b> area Russia world ethnic Japan present Myanmar people most policy India Asia Kokang center speak South-Korea Western	USA <b>nation</b> China world-wide area India border Russia union Japan phenomenon center policy ethnic western controversy small speaker present negotiation	China USA people <b>nation</b> Japan world mainland-China policy present Russia ethnic small Taiwan human worldwide western awareness India first democracy

Each row lists the top 20 words with highest probabilities under the topics

comments of news articles, as users who hold opposite viewpoints firstly argue with each other and then the argument could evolve into mutual verbal abuses. However, the correlation coefficients of word probabilities ( $\phi$ ) between these two topics and topics extracted by BTM-B and PTM-B are extremely low, indicating that these two topics have not been discovered by the other competitors.

### 4.2.3 Overall evaluation of topics

After separate evaluations of formal and informal topics, we use an automated metric, coherence score, to evaluate the overall quality of topics learned by COTM. In the experiments, the coherence score of both formal and informal topics learned by COTM is compared with the scores of topics learned by LDA-P, BTM-B, PTM-B and EXTM. We do not show coherence scores for LCTM-P, since the topics extracted by this model are distributions over concepts, not words. The topic numbers for COTM are set to be  $K_{COTM} = 100$  and  $J_{COTM} = 20, 40, 80$ , separately. The numbers of master topics and extended topics in EXTM are the same with those for COTM. The corresponding topic numbers for LDA-P, BTM-B and PTM-B are set to be 120, 140, 180 correspondingly. From Table 9, we find that COTM and PTM-B achieve comparable results, with PTM-B outperforming COTM when the topic number is large (180) and COTM achieving slightly better results when the topic numbers are small (120 and 140). Besides, both PTM-B and COTM consistently outperform the other three models in all experimental settings.

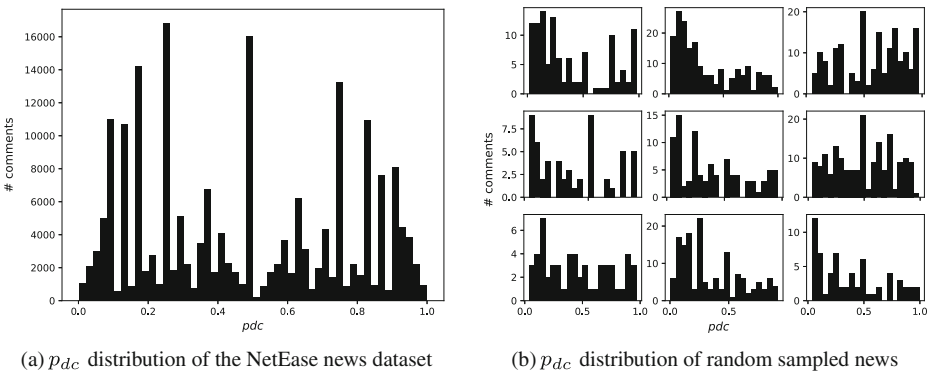
**Table 8** Unique informal topics discovered by COTM from the NetEase dataset

Rude	Talking	Whole-family die man curse wife die-out generations be-slaves sister mother eunuch sucks stink widow unable-to-die-a-natural-death fuck dick ass speak beast
Mutual	Judgement	Sucks stupid second-floor freaking-awesome first-floor since-antiquity dog curse upstairs die mother-fucker funny nice sucker pretentious-bastard talking-big know support monkey reply

**Table 9** Average coherence scores of topics learned by COTM and its competitors. A larger coherence score indicates more coherence topics

# All Topics		120			140			180		
Data	Method	Top5	Top10	Top20	Top5	Top10	Top20	Top5	Top10	Top20
NetE	LDA-P	-9.72	-60.63	-296.07	-9.75	-60.88	-293.93	-9.44	-59.73	-288.06
	BTM-B	-9.49	-59.49	-302.91	-10.37	-59.20	-318.12	-10.32	-59.04	-312.55
	PTM-B	-8.96	-55.62	-269.51	-8.81	-54.80	<b>-266.28</b>	<b>-7.53</b>	<b>-50.28</b>	<b>-258.70</b>
	EXTM	-9.12	-58.13	-285.76	-9.25	-59.14	-290.02	-9.31	-59.27	-289.06
	COTM	<b>-8.37</b>	<b>-53.78</b>	<b>-268.91</b>	<b>-8.72</b>	<b>-54.30</b>	-270.17	-8.69	-55.41	-272.93
	COTM-F	<b>-7.93</b>	<b>-45.65</b>	<b>-221.03</b>	<b>-8.01</b>	<b>-47.41</b>	<b>-225.00</b>	-8.87	-50.50	<b>-227.90</b>
	COTM-I	-8.73	-58.44	-278.17	-9.39	-56.86	-273.29	-9.00	-56.02	-279.67
Sina	LDA-P	-9.01	-57.65	-280.96	-9.83	-56.95	-290.74	-10.33	-59.36	-298.26
	BTM-B	-8.66	-53.16	-257.58	-9.83	-55.47	-280.93	-9.42	-53.08	-296.82
	PTM-B	-8.62	-52.67	-255.97	<b>-9.26</b>	-55.15	-274.90	<b>-8.22</b>	<b>-50.90</b>	<b>-251.47</b>
	EXTM	-8.65	-52.96	-256.03	-9.57	-55.42	-278.26	-9.13	-54.87	-282.34
	COTM	<b>-8.59</b>	<b>-52.43</b>	<b>-248.19</b>	-9.27	<b>-54.90</b>	<b>-271.37</b>	-8.91	-54.32	-274.43
	COTM-F	<b>-7.19</b>	<b>-44.64</b>	<b>-200.37</b>	<b>-7.53</b>	<b>-47.98</b>	<b>-223.15</b>	<b>-7.40</b>	<b>-50.72</b>	<b>-225.21</b>
	COTM-I	-8.63	-52.90	-251.63	-9.83	-55.38	-272.07	-9.13	-54.63	-278.74

To further explore the quality of formal topics and informal topics learned by COTM, we calculate the average coherence scores only on formal topics and informal topics respectively, the results of which are defined as COTM-F and COTM-I in Table 9. We find that the average coherence scores of formal topics are higher than those of informal topics, and even become the highest in nearly all experimental settings. These findings indicate that COTM shows strong performances in learning formal topics and poor performances in learning informal topics, and therefore achieves comparable performances with PTM-B in modeling both of the normal documents and short texts. This phenomenon validates the basic assumption of COTM that borrowing external information from short texts can help improve the topic learning of formal topics. As for informal topics, since they are only formed by short texts which are not much correlated with normal documents, the less amount of content information results in their poor performance.



**Figure 5**  $p_{dc}$  distribution of NetEase news dataset and random sampled pieces of NetEase news, with  $K_{COTM} = 100$  and  $J_{COTM} = 20$

#### 4.2.4 Detection of spam short texts

In recent years, detecting spam reviews on the Web has gained great importance. The relevant text corpora often consists of co-occurring normal documents and following short texts. For instance, products or services are often described by normal textual introductions on the electronic commerce website, with each product description followed by a number of short buyers’ reviews. Spam detection can help filter out “untruthful reviews”, “brands reviews” and “non-reviews” for products or services [7], which are highly concerned by manufacturers and retailers [6].

Exploring the topical relationships between normal documents and short texts also sheds light on detecting spams. For example, EXTM uses a switch variable  $H$  for each short text to decide whether it talks about a slave topic or an extended topic. Short texts that are classified as discussing extended topics can be regarded as spams. However, in real situations such as buyers’ reviews, short texts may not only talk about slave topics derived from normal product descriptions, but also include additional personal opinions. In this scenario, using

**Table 10** A sampled news article and its corresponding user comments

news article			
Daily-Mail report Vietnam Kite Festival year old boy kite rope grab hold kite drag air figure at-scene boy dragged into air giant kite boy drag sloshing around finally fall down afterward emergency send hospital rescue boy kite accident happen moment ChinaNews foreign-media report Vietnam Ho-Chi-Minh-City days-ago take-place kite-flying result in boy fall dead kite meters-wide local kite association take-off rope accidentally rapped boy fly-into-air finally tragically fall dead belong-to local association trying fly kite boy suddenly approach staff concentrated fly kite notice currently Vietnam government investigate original title Vietnam years-old boy accidentally giant kite take air fall dead			
User comments		$p_{dc}$	$1 - p_{dc}$
Relevant	no-matter reason lead child dead tragedy forty thumbs-up ruthless maybe lost kinsfolk pain sober	0.97368	0.02632
	child dreamt kite fly-into-sky	0.90000	0.10000
	real tragic boy	0.87500	0.12500
	unreal must real high wind blow away fall dead	0.72222	0.27778
	read more common sense learned know kite wide meters wind power blow away people young child	0.53571	0.46429
	real fly sky	0.51287	0.48713
	Irrelevant	Wangfeng should have everyday sing fly higher	0.16667
stand by first commenter		0.16667	0.83333
first commenter stupid		0.12500	0.87500
people Shandong province less go abroad cursed		0.10000	0.90000
Mi phone outstanding empty-talk devastated		0.01087	0.98913
all staff hardworking exhausted happiness Chinese no reason support domestic			

**Table 11** Average coherence scores of topics learned by oLDA-B and oCOTM

Number of Topics		120			140			180		
Data	Method	Top5	Top10	Top20	Top5	Top10	Top20	Top5	Top10	Top20
NetE	oLDA-B	-9.82	-61.54	-298.44	-9.86	-61.74	-295.68	-9.74	-60.22	-290.35
	oCOTM	<b>-8.54</b>	<b>-56.01</b>	<b>-270.93</b>	<b>-8.63</b>	<b>-55.87</b>	<b>-268.50</b>	-8.06	<b>-51.99</b>	<b>-258.82</b>
Sina	oLDA-B	-8.96	-55.94	-277.82	-9.12	-55.96	-286.58	-9.30	-57.89	-289.65
	oCOTM	<b>-8.17</b>	<b>-51.59</b>	<b>-243.38</b>	<b>-8.13</b>	<b>-51.92</b>	<b>-260.36</b>	-8.22	<b>-50.90</b>	<b>-251.47</b>

switch variables to simply classify short texts into two groups is an assumption that is too strong for modeling the topical meanings of short texts.

In a more natural way, the COTM model uses association probabilities  $p_{dc}$  to describe topical relationships of normal documents and their corresponding short texts. As is shown in Figure 5, the association probabilities between NetEase news and their following short reader comments vary a lot between 0 and 1. Figure 5a shows the distribution of  $p_{dc}$  for the entire corpus and we find several obvious peaks within the interval. Figure 5b presents the  $p_{dc}$  distributions of 9 randomly sampled NetEase news and their corresponding reader comments, which shows different patterns. As a result, we can draw a conclude that short texts have different topical relationships with their co-occurring normal documents. So we propose to use  $p_{dc}$  in COTM to automatically detect potential spam texts. Specifically, short texts with  $p_{dc}$  less than a certain threshold  $\tilde{p}$  are not much semantically related to their corresponding normal documents, and thus can be classified as irrelevant short texts, which are often spams.

To illustrate the ability of using  $p_{dc}$  to detect spams, the top half of Table 10 shows one sampled news article reporting an accident that took place in Vietnam, and the second half shows sampled user comments following the news article. All sampled comments are sorted by  $p_{dc}$  in descending order, and are further classified into relevant and irrelevant comments, with the threshold being  $\tilde{p} = 0.2$ . It can be observed that relevant comments talk about the original article and express preaches and feelings of sorrow, surprise or ridicule. On the contrary, irrelevant comments could be random talks, verbal abuses or advertising. This

**Table 12** Average coherence score of topics learned by online BTM algorithms and informal topics learned by oCOTM

J		20			40			80		
Data	Method	Top5	Top10	Top20	Top5	Top10	Top20	Top5	Top10	Top20
NetE	oBTM-S	-10.33	-61.70	-304.37	-9.97	-60.83	-297.83	-9.74	-61.85	-301.24
	iBTM-S	-11.06	-62.40	-300.47	-10.55	-61.63	-297.18	-10.72	-63.00	-308.00
	oCOTM	<b>-10.25</b>	<b>-60.41</b>	<b>-300.70</b>	<b>-9.73</b>	<b>-57.96</b>	<b>-279.85</b>	<b>-9.04</b>	<b>-53.51</b>	<b>-262.18</b>
Sina	oBTM-S	-8.92	-53.03	-260.86	-9.53	-55.57	-284.78	-9.11	-58.01	-301.62
	iBTM-S	-9.11	-53.21	-257.91	-9.67	-55.82	-289.14	-9.15	-59.36	-307.52
	oCOTM	<b>-8.89</b>	<b>-52.67</b>	<b>-248.48</b>	<b>-9.50</b>	<b>-54.56</b>	<b>-269.41</b>	-8.58	<b>-53.76</b>	<b>-250.81</b>

example demonstrate that COTM can be efficiently used for identifying topically irrelevant comments, and can be potentially used to detect spam reviews.

### 4.3 Evaluation of online COTM

#### 4.3.1 Topic coherence

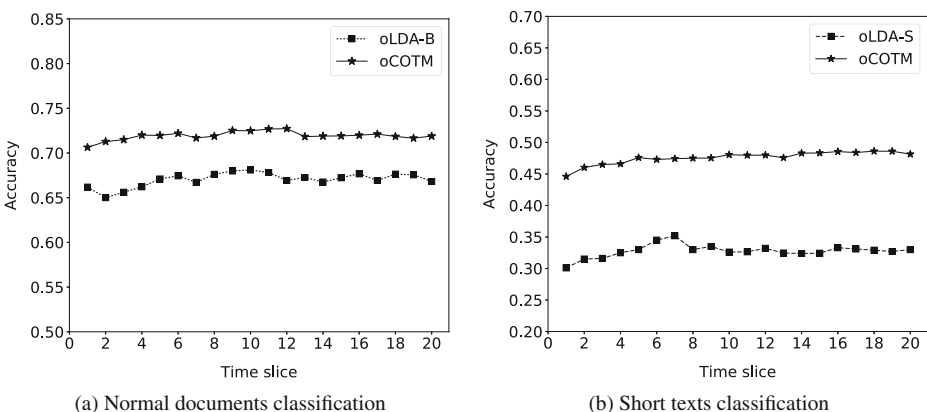
To evaluate the quality of topics learned by online algorithms, we compare the average coherence scores of the models. We first compare oCOTM with oLDA-B, and set  $K_{oCOTM} = 100$ ,  $K_{oLDA-B} = K_{oCOTM} + J_{oCOTM}$  where  $J_{oCOTM}$  is 20, 40 or 80. From the coherence scores shown in Table 11, we see that oCOTM achieves higher topic coherence scores than oLDA-B in all experimental settings for the two datasets.

We next use average coherence scores to compare the quality of informal topics learned by oCOTM with the quality of topics learned by oBTM-S and iBTM-S. We set  $K_{oBTM-S} = K_{iBTM-S} = J_{oCOTM}$ . While online BTM algorithms use biterns from the entire collection of short texts, the learning of informal topics in oCOTM only gains knowledge from a subset of words in each short comment. This difference implies an additional inherent advantage to the BTM algorithms, besides the one mentioned before that BTM's basic assumption is consistent with the definition of coherent score. Nonetheless, we still find that oCOTM outperforms iBTM-S and oBTM-S in all experimental settings for the two datasets, as shown in Table 12.

#### 4.3.2 Document classification

We further evaluate the quality of topic representation of documents learned by oCOTM through document classification for the Sina dataset.

In the experiment of classifying normal documents, we set  $K_{oCOTM} = 100$ ,  $J_{oCOTM} = 50$  and  $K_{oLDA-B} = 150$ . Topic proportions of normal documents are used as features in classification. From the results shown in Figure 6a, we observe that the accuracy of oCOTM is higher than oLDA-B. In the experiment of classifying short texts, we set  $K_{oCOTM} = 100$ , and  $J_{oCOTM} = K_{oLDA-S} = 50$ . From the results shown in Figure 6b, we find that oCOTM outperforms oLDA-S dramatically. Overall, oCOTM achieves the best performance both in classifying normal documents and short texts at all time slices.



**Figure 6** Comparison of classification performance of online algorithms on Sina dataset

## 5 Conclusion

With the development of online services, co-occurring normal documents and short texts are becoming increasingly prevalent throughout the Internet. Conventional topic models designed for normal texts or short texts are not applicable to these texts with co-occurring structure. In this paper, we propose a novel topic model, namely COTM, to deal with this kind of text corpora. The COTM model can directly exploit the co-occurring structure, and use information from both normal documents and short texts to learn topics in a mutually reinforced way. We also introduce an online algorithm for COTM, referred to as oCOTM, to deal with large scale datasets. Extensive experiments on the NetEase news and Sina blog datasets demonstrate that COTM outperforms several state-of-art models in various ways, including learning more prominent and comprehensive topics, and getting better topic representations of documents. Besides, COTM can be potentially used for unsupervised detection of spam reviews.

**Acknowledgements** This work is funded by the State Key Development Program of Basic Research of China (973) under Grant No. 2013cb329600 and National Natural Science Foundation of China under Grant Nos. 61672050, 61372191, 61472433, 61572492.

## Appendix: Details of deriving the collapsed gibbs sampling algorithm

Given the full posterior distribution in (1), we can easily get the full conditional posterior distributions for  $\Theta$ ,  $\Phi$ ,  $\Psi$ ,  $\xi$  and  $P$ .

For  $\theta_d, d \in \{1, 2, \dots, D\}$ , its full conditional posterior distribution is:

$$f(\theta_d | \cdot) \propto \prod_{k=1}^K (\theta_{dk})^{l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha - 1}. \tag{17}$$

For  $\phi_k, k \in \{1, 2, \dots, K\}$ , its full conditional posterior distribution is:

$$f(\phi_k | \cdot) \propto \prod_{v=1}^V (\phi_{kv})^{l_{kv}^{(2)} + g_{kv}^{(2)} + \beta - 1}. \tag{18}$$

For  $\psi_j, j \in \{1, 2, \dots, J\}$ , its full conditional posterior distribution is:

$$f(\psi_j | \cdot) \propto \prod_{v=1}^V (\psi_{jv})^{g_{jv}^{(3)} + \beta - 1}. \tag{19}$$

For  $\xi$ , its full conditional posterior distribution is:

$$f(\xi | \cdot) \propto \prod_{j=1}^J (\epsilon_j)^{h_j + \epsilon - 1}. \tag{20}$$

For  $p_{dc}, d \in \{1, 2, \dots, D\}, c \in \{1, 2, \dots, C_d\}$ , its full conditional posterior distribution is:

$$f(p_{dc} | \cdot) \propto (p_{dc})^{s_{dc}^{(1)} + \gamma - 1} (1 - p_{dc})^{s_{dc}^{(2)} + \gamma - 1}. \tag{21}$$

Noting the posterior distributions of  $\Theta, \Phi, \Psi, \xi$  and  $P$  are all Dirichlet, conjugate with their priors, we can develop a collapsed Gibbs sampling algorithm by integrating out these parameters from the posterior distribution.



To describe this procedure, we start with introducing the Dirichlet distribution. Suppose  $\mathbf{X} = (X_1, \dots, X_K)^T$ , following a Dirichlet distribution with parameter  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)^T$ . The probability density function of  $\mathbf{X}$  is

$$f(\mathbf{X}|\boldsymbol{\alpha}) = f(X_1, \dots, X_K|\alpha_1, \dots, \alpha_K) = \frac{\Gamma\left(\sum_{i=1}^K \alpha_i\right)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K X_i^{\alpha_i-1}. \tag{22}$$

Since the integral of  $f(\mathbf{X}|\boldsymbol{\alpha})$  is equal to 1, we can get

$$\int \left\{ \prod_{i=1}^K X_i^{\alpha_i-1} \right\} dX_1 \dots dX_K = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^K \alpha_i\right)}. \tag{23}$$

Similarly, given the conditional posterior distribution of  $\Theta$  is Dirichlet, as described in (17), we can integrate it out and get:

$$\begin{aligned} & \int f(\Theta | \cdot) d\Theta = \prod_{d=1}^D \int f(\theta_d | \cdot) d\theta_d \\ & \propto \prod_{d=1}^D \int \left\{ \prod_{k=1}^K (\theta_{dk})^{l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha - 1} \right\} d\theta_{d1} \dots d\theta_{dK} = \prod_{d=1}^D \frac{\prod_{i=1}^K \Gamma\left(l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha\right)}{\Gamma\left\{\sum_{k=1}^K (l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha)\right\}}. \end{aligned} \tag{24}$$

Then we integrate out  $\Phi$ ,  $\Psi$ ,  $\xi$  and  $\mathbf{P}$  similarly and get the following results:

$$\begin{aligned} \int f(\Phi | \cdot) d\Phi &= \prod_{k=1}^K \int f(\phi_k | \cdot) d\phi_k \propto \prod_{k=1}^K \frac{\prod_{v=1}^V \Gamma\left(l_{kv}^{(2)} + g_{kv}^{(2)} + \beta\right)}{\Gamma\left\{\sum_{v=1}^V (l_{kv}^{(2)} + g_{kv}^{(2)} + \beta)\right\}} \\ \int f(\Psi | \cdot) d\Psi &= \prod_{j=1}^J \int f(\psi_j | \cdot) d\psi_j \propto \prod_{j=1}^J \frac{\prod_{v=1}^V \Gamma\left(g_{jv}^{(3)} + \beta\right)}{\Gamma\left\{\sum_{v=1}^V (g_{jv}^{(3)} + \beta)\right\}} \\ \int f(\xi | \cdot) d\xi &\propto \frac{\prod_{j=1}^J \Gamma(h_j + \epsilon)}{\Gamma\left\{\sum_{j=1}^J (h_j + \epsilon)\right\}} \\ \int f(\mathbf{P} | \cdot) d\mathbf{P} &= \prod_{d=1}^D \prod_{c=1}^{C_d} \int f(p_{dc} | \cdot) dp_{dc} \propto \prod_{d=1}^D \prod_{c=1}^{C_d} \frac{\Gamma\left(s_{dc}^{(1)} + s_{dc}^{(2)} + \gamma\right)}{\Gamma\left(s_{dc}^{(1)} + \gamma\right) \Gamma\left(s_{dc}^{(2)} + \gamma\right)} \end{aligned} \tag{25}$$

By integrating out  $\Theta$ ,  $\Phi$ ,  $\Psi$ ,  $\xi$  and  $\mathbf{P}$ , the full posterior distribution in (1) can be simplified as:

$$\begin{aligned} & f(\mathbf{z}, \mathbf{b}, \mathbf{x}, \mathbf{y} | \mathbf{w}, \boldsymbol{\alpha}, \beta, \gamma, \epsilon) \\ &= \int f(\mathbf{z}, \mathbf{b}, \mathbf{P}, \mathbf{x}, \mathbf{y}, \Theta, \Phi, \Psi, \xi | \mathbf{w}, \boldsymbol{\alpha}, \beta, \gamma, \epsilon) d\Theta d\Phi d\Psi d\xi d\mathbf{P} \\ &\propto \prod_{d=1}^D \frac{\prod_{i=1}^K \Gamma\left(l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha\right)}{\Gamma\left\{\sum_{k=1}^K (l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha)\right\}} \times \prod_{k=1}^K \frac{\prod_{v=1}^V \Gamma\left(l_{kv}^{(2)} + g_{kv}^{(2)} + \beta\right)}{\Gamma\left\{\sum_{v=1}^V (l_{kv}^{(2)} + g_{kv}^{(2)} + \beta)\right\}} \\ &\times \prod_{j=1}^J \frac{\prod_{v=1}^V \Gamma\left(g_{jv}^{(3)} + \beta\right)}{\Gamma\left\{\sum_{v=1}^V (g_{jv}^{(3)} + \beta)\right\}} \times \frac{\prod_{j=1}^J \Gamma(h_j + \epsilon)}{\Gamma\left\{\sum_{j=1}^J (h_j + \epsilon)\right\}} \times \prod_{d=1}^D \prod_{c=1}^{C_d} \frac{\Gamma\left(s_{dc}^{(1)} + s_{dc}^{(2)} + \gamma\right)}{\Gamma\left(s_{dc}^{(1)} + \gamma\right) \Gamma\left(s_{dc}^{(2)} + \gamma\right)}. \end{aligned} \tag{26}$$

Thus, we can use the collapsed Gibbs sampling and only need to update  $\mathbf{z}$ ,  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{b}$  in each iteration. We then derive the conditional posterior distributions of  $\mathbf{z}$ ,  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{b}$  from (26).

Specifically, for the  $n$ th word in normal document  $d$ ,  $z_{dn} = k$  only influences  $l_{dk}^{(1)}$  and  $l_{kw_{dn}}^{(2)}$  in (26). Let  $\mathbf{z}_{-dn}$  denote  $\mathbf{z}$  excluding  $z_{dn}$ , and the full conditional distribution of  $z_{dn}$  can be derived as:

$$\begin{aligned}
 f(z_{dn} = k | \cdot) &= \frac{f(z_{dn} = k, \mathbf{z}_{-dn} | \cdot)}{f(\mathbf{z}_{-dn} | \cdot)} \\
 &\propto \frac{\Gamma(l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha)}{\Gamma(l_{dk;-dn}^{(1)} + g_{dk}^{(1)} + \alpha)} \frac{\Gamma\left\{\sum_{k' \neq k} (l_{dk'}^{(1)} + g_{dk'}^{(1)} + \alpha) + (l_{dk;-dn}^{(1)} + g_{dk}^{(1)} + \alpha)\right\}}{\Gamma\left\{\sum_{k' \neq k} (l_{dk'}^{(1)} + g_{dk'}^{(1)} + \alpha) + (l_{dk}^{(1)} + g_{dk}^{(1)} + \alpha)\right\}} \\
 &\times \frac{\Gamma(l_{kw_{dn}}^{(2)} + g_{kw_{dn}}^{(2)} + \beta)}{\Gamma(l_{kw_{dn};-dn}^{(2)} + g_{kw_{dn}}^{(2)} + \beta)} \frac{\Gamma\left\{\sum_{v \neq w_{dn}} (l_{kv}^{(2)} + g_{kv}^{(2)} + \beta) + (l_{kw_{dn};-dn}^{(2)} + g_{kw_{dn}}^{(2)} + \beta)\right\}}{\Gamma\left\{\sum_{v \neq w_{dn}} (l_{kv}^{(2)} + g_{kv}^{(2)} + \beta) + (l_{kw_{dn}}^{(2)} + g_{kw_{dn}}^{(2)} + \beta)\right\}} \\
 &\propto \frac{\Gamma(l_{dk;-dn}^{(1)} + 1 + g_{dk}^{(1)} + \alpha)}{\Gamma(l_{dk;-dn}^{(1)} + g_{dk}^{(1)} + \alpha)} \frac{\Gamma(l_{d;-dn}^{(1)} + g_{d.}^{(1)} + K\alpha)}{\Gamma(l_{d;-dn}^{(1)} + 1 + g_{d.}^{(1)} + K\alpha)} \\
 &\times \frac{\Gamma(l_{kw_{dn};-dn}^{(2)} + 1 + g_{kw_{dn}}^{(2)} + \beta)}{\Gamma(l_{kw_{dn};-dn}^{(2)} + g_{kw_{dn}}^{(2)} + \beta)} \frac{\Gamma(l_{k;-dn}^{(2)} + g_{k.}^{(2)} + V\beta)}{\Gamma(l_{k;-dn}^{(2)} + 1 + g_{k.}^{(2)} + V\beta)}, \tag{27}
 \end{aligned}$$

where the subscript “ $-dn$ ” indicates counts excluding the  $n$ th word in normal document  $d$ ,  $l_{d.}^{(1)}$  and  $g_{d.}^{(1)}$  are the sum of  $l_{dk}^{(1)}$  and  $g_{dk}^{(1)}$  over all formal topics  $k$ , and  $l_{k.}^{(2)}$  and  $g_{k.}^{(2)}$  are the sum of  $l_{kv}^{(2)}$  and  $g_{kv}^{(2)}$  over all words  $v$ . Noting  $l_{d.}^{(1)}$  is equal to the total number of words in document  $d$  and  $g_{d.}^{(1)}$  is equal to the total number of words in all short texts associated with normal document  $d$ ,  $l_{d.}^{(1)}$  and  $g_{d.}^{(1)}$  are constant values. Using the characteristics of  $\Gamma$  function, which is  $\Gamma(x + 1) = x\Gamma(x)$ , (27) can be simplified as

$$f(z_{dn} = k | \cdot) \propto (l_{dk;-dn}^{(1)} + g_{dk}^{(1)} + \alpha) \times \frac{l_{k,w_{dn};-dn}^{(2)} + g_{k,w_{dn}}^{(2)} + \beta}{l_{k;-dn}^{(2)} + g_{k.}^{(2)} + V\beta}.$$

For  $\mathbf{b}$ ,  $\mathbf{x}$ ,  $\mathbf{y}$ , we can derive their conditional posterior distributions from (26) similarly.

## References

1. AlSumait, L., Barbara, D., Domeniconi, C.: On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In: 2008 eighth IEEE international conference on data mining, pp. 3c12. IEEE (2008)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993C1022 (2003)
3. Cai, D., Mei, Q., Han, J., Zhai, C.: Modeling hidden topics on document manifold. In: Proceedings of the 17th ACM conference on information and knowledge management, pp. 911c920. ACM (2008)
4. Calinski, T., Harabasz, J.: A dendrite method for cluster analysis. *Communications in Statistictheory and Methods* **3**(1), 1C27 (1974)
5. Cheng, X., Yan, X., Lan, Y., Guo, J.: Btm: Topic modeling over short texts. *IEEE Trans. Knowl. Data Eng.* **26**(12), 2928C2941 (2014)
6. Crawford, M., Khoshgoftaar, T.M., Prusa, J.D., Richter, A.N., Al Najada, H.: Survey of review spam detection using machine learning techniques. *J. Big Data* **2**(1), 1C24 (2015)
7. Dixit, S., Agrawal, A.: Survey on review spam detection. *Int. J. Comput. Commun. Technol. ISSN (PRINT)* **4**, 0975C7449 (2013)

8. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. *J. Mach. Learn. Res.* **9**(Aug), 1871C1874 (2008)
9. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, pp. 50c57. ACM (1999)
10. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics, pp. 80c88. ACM (2010)
11. Hu, W., Tsujii, J.: A latent concept topic model for robust topic inference using word embeddings. In: The 54th annual meeting of the association for computational linguistics, pp. 380 (2016)
12. Jin, O., Liu, N.N., Zhao, K., Yu, Y., Yang, Q.: Transferring topical knowledge from auxiliary long texts for short text clustering. In: Proceedings of the 20th ACM international conference on information and knowledge management, pp. 775c784. ACM (2011)
13. Lakkaraju, H., Bhattacharya, I., Bhattacharyya, C.: Dynamic multi-relational chinese restaurant process for analyzing influences on users in social media. In: 2012 IEEE 12th international conference on data mining, pp. 389c398. IEEE (2012)
14. Li, C., Wang, H., Zhang, Z., Sun, A., Ma, Z.: Topic modeling for short texts with auxiliary word embeddings. In: The international ACM SIGIR conference, pp. 165c174 (2016)
15. Liu, Y., Niculescu-Mizil, A., Gryc, W.: Topic-link lda: joint models of topic and author community. In: Proceedings of the 26th annual international conference on machine learning, pp. 665c672. ACM (2009)
16. Ma, Z., Sun, A., Yuan, Q., Cong, G.: Topic-driven reader comments summarization. In: Proceedings of the 21st ACM international conference on information and knowledge management, pp. 265c274. ACM (2012)
17. McCallum, A., Wang, X., Mohanty, N.: Joint group and topic discovery from relations and text. Springer (2007)
18. Mehrotra, R., Sanner, S., Buntine, W., Xie, L.: Improving lda topic models for microblogs via tweet pooling and automatic labeling. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, pp. 889c892. ACM (2013)
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *Computer Science* (2013)
20. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proceedings of the conference on empirical methods in natural language processing, association for computational linguistics, pp. 262c272 (2011)
21. Natarajan, N., Sen, P., Chaoji, V.: Community detection in content-sharing social networks. In: Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining, pp. 82c89. ACM (2013)
22. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Conference on empirical methods in natural language processing, pp. 1532c1543 (2014)
23. Phan, X.H., Nguyen, L.M., Horiguchi, S.: 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, pp. 91c100. ACM (2008)
24. Phan, X.H., Nguyen, C.T., Le, D.T., Nguyen, L.M., Horiguchi, S., Ha, Q.T.: A hidden topic-based framework toward building applications with short Web documents. *IEEE Trans. Knowl. Data Eng.* **23**(7), 961C976 (2011)
25. Quan, X., Kit, C., Ge, Y., Pan, S.J.: Short and sparse text topic modeling via self-aggregation. In: International conference on artificial intelligence, pp. 2270c2276 (2015)
26. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterank: finding topic-sensitive influential twitterers. In: Proceedings of the third ACM international conference on Web search and data mining, pp. 261c270. ACM (2010)
27. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: Proceedings of the 22nd international conference on WorldWideWeb, InternationalWorldWideWeb conferences steering committee, pp. 1445c1456 (2013)
28. Yang, Y., Wang, F., Jiang, F., Jin, S., Xu, J.: A topic model for hierarchical documents. In: International conference on data science in cyberspace, IEEE (2016)
29. Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing twitter and traditional media using topic models. In: Advances in information retrieval, pp. 338c349. Springer (2011)
30. Zuo, Y., Wu, J., Zhang, H., Lin, H., Wang, F., Xu, K., Xiong, H.: Topic modeling of short texts: a pseudo-document view. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 2. ACM (2016)