Information Sciences 456 (2018) 83–96

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Filtering out the noise in short text topic modeling

Ximing Li[a,b], Yue Wang[a,b], Ang Zhang[a,b], Changchun Li[a,b], Jinjin Chi[a,b], Jihong Ouyang[a,b,*]

[a] College of Computer Science and Technology, Jilin University, China
[b] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China

ARTICLE INFO

Article history:
Received 23 March 2017
Revised 21 April 2018
Accepted 25 April 2018
Available online 3 May 2018

Keywords:
Short text
Topic modeling
Noise words
Topic inference

ABSTRACT

Nowadays, massive short texts, such as social media posts and newspaper titles, are available on the Internet. Analyzing these short texts is very significant for many content analysis tasks. However, the commonly used text analysis tools, i.e., topic models, lose effectiveness on short texts because of the sparsity and noise problems. Recent topic models mainly attempt to solve the sparsity problem, but neglect the noise issue. To address this, we propose a common semantics topic model (CSTM) in this paper. The key idea is to introduce a new type of topic, namely common topic, to gather the noise words. The experimental results on real-world datasets indicate that our CSTM outperforms the existing short text topic models on the traditional tasks.

1. Introduction

Short texts from social media, such as tweets and Q&A websites, are increasingly available on the Internet, and they are used in a wide range of content analysis tasks, e.g., online advertising and query suggestion. However, short texts are commonly sparse and noisy, resulting in severe sparsity and noise problems. To illustrate this, we present two real-world short texts from tweets after removing the standard stop words:

1. *{various ways generate website traffic rt}*
2. *{haha funny story hahaha}*

The document length of the first text is six, and that of the second text is only four. Additionally, both of them contain noise words such as “rt” (i.e., the abbreviation of “retweet”), “haha” and “hahaha”. Generally, these two problems bring significant challenges to short text mining tasks.

In this paper, we are interested in short text topic modeling, which also suffers from the sparsity and noise problems mentioned above [8,28]. Topic models, such as latent Dirichlet allocation (LDA) [3] and correlated topic model [2], assume that each document is described by a distribution over topics, where each topic is a distribution over a fixed vocabulary. The statistical inference algorithms are used to train the distributions with respect to topics. However, too few and noisy word tokens (i.e., samples) lead to inaccurate estimations of topic distributions at the document level.

A straightforward way to improve short text topic modeling is to aggregate short texts into long pseudo-documents before training a traditional topic model [8,12,14,20,23,29]. Some attempts aggregated short texts of tweets using the user

* Corresponding author.
E-mail addresses: liximing86@gmail.com, ouyj@jlu.edu.cn (J. Ouyang).

https://doi.org/10.1016/j.ins.2018.04.071
0020-0255/

information [8], shared words [23] and combinations of various side messages [14]. Besides these methods that are highly data-dependent, some algorithms are designed to be adaptively aggregated short texts. For example, the authors of Quan et al. [20] integrated topic modeling with clustering, and then proposed an adaptive aggregation topic model (SATM) for short texts. However, the adaptive short text aggregation step of SATM is computationally expensive, especially for collections of massive short texts.

Another way is to investigate generalized topic models for short texts by refining model structures. A simple but effective model is the mixture of unigrams [18], which posits that each document is drawn from a single topic. This single topic assumption indirectly alleviates the sparsity problem at the document level [9,28]. Some recent algorithms used global word co-occurrence information to alleviate the sparsity problem of short texts. The biterm topic model (BTM) proposed in [6,26] directly models word co-occurrence patterns (i.e., biterms) over the whole corpus. Zuo et al. [30] modeled short texts by constructing a global word co-occurrence network. Additionally, the authors of Sridhar [22] integrated topic modeling with word embeddings, and then assumed that a short text corpus is a mixture of Gaussians on the embedding space. Empirical results showed that these models worked well on short texts, however, they neglect the noise problem, which is also crucial for short text topic modeling. Taking BTM as an example, the base unit biterm in BTM is defined as an unordered word co-occurrence pattern, and any two distinct words in a short text construct a biterm. If there is a noise word, it must be a partner with any other words in this short text, resulting in many low-quality biterms.

This paper aims at developing a generalized short text topic model that further solves the noise problem. Roughly, we consider two most common types of noise words. The first type is the high frequency domain-specific noise word. A prefect example is the word “rt”, which is (almost) only occurring in Twitter posts but has extremely high frequency. The second type is the word in colloquial style, such as “haha”, “hahaha” and “hahahaha”. It is very difficult to capture the links between them. Worse of all, the occurrence frequencies of these noise words (e.g., “haha”) are neither very low nor high, but just like the occurrence frequencies of informative words. You cannot detect them by counting their occurrence numbers. Overall speaking, we argue that filtering out such noise words in short texts is not easy. We cannot use traditional document preprocessing methods to effectively remove the noise of short texts.

We attempt to solve the noise problem by adding a generative process of noise words into the model structure, and then develop a novel common semantics topic model (CSTM). In CSTM, we define a new type of topic, namely common topic, to capture common semantics and noise words. In contrast, the conventional topics that describe specific subjects are named function topics. Basically, CSTM is built on the mixture of unigrams model. We assume that each short text is a mixture of a selected function topic and all common topics. Common topics prefer background words and noise words, and then help filtering out such noise. To evaluate the performance of CSTM, we conducted extensive experiments on three real-world short text collections. The experimental results indicate that compared to the traditional short topic models, CSTM can discover more coherent topics (i.e., measured by NPMI [4]), and it performs better on the classification and clustering tasks.

The rest of this paper is organized as follows: In Section 2, we review related works and basic models. Section 3 presents the proposed CSTM model. The experimental results are shown in Section 4. In Section 5, we present the conclusion and potential future works.

2. Background

In this section, we first review some related works, and then introduce the mixture of unigrams model, which is the basic model of our CSTM. To better describe the noise problem, we also give a brief introduction to another generalized short text topic model, i.e., BTM.

2.1. Related work

The researchers developed some existing generalized short text topic models. A simple but effective topic model for short text is the mixture of unigrams [18,27] model. Basically, it can indirectly alleviate the sparsity problem using the single topic assumption at the document level [9,28]. The recent BTM [6,26] uses global word co-occurrences (i.e., biterms), and directly models biterms over the whole corpus. Besides mixture of unigrams and BTM, another recent generalized topic model for short text is the word network topic model (WNTM) proposed in [30]. WNTM regroups a short text corpus by constructing a global word co-occurrence network. Each word type is treated as a pseudo-document with content consisting of its adjacent word types in the network. Most of these word type pseudo-documents are lengthy, enriching the word token samples at the pseudo-document level. These models can alleviate the sparsity problem in some degree, but they neglect the noise problem. For BTM, the noise words may result in many low-quality biterms; and for WNTM, word type pseudo-documents may be filled with background noise words, which are frequently co-occurring with most of word types, e.g., the word “rt” in Twitter posts. Additionally, WNTM is time-consuming with large global word co-occurrence network.

There exist some previous models using a background topic to distill discriminative words in tweets. In Twitter-LDA [28], word tokens can be sampled from the standard topic or a background topic. Twitter-TTM [21] improves Twitter-LDA by further estimating the ratio of the background topic to standard topics for each user. Twitter-BTM [5], an extension of BTM, also adds a background topic into the generative process of documents. These models can use the background topic to filter out background noise words in some degree, however, they are dependent on the available user information in tweets.

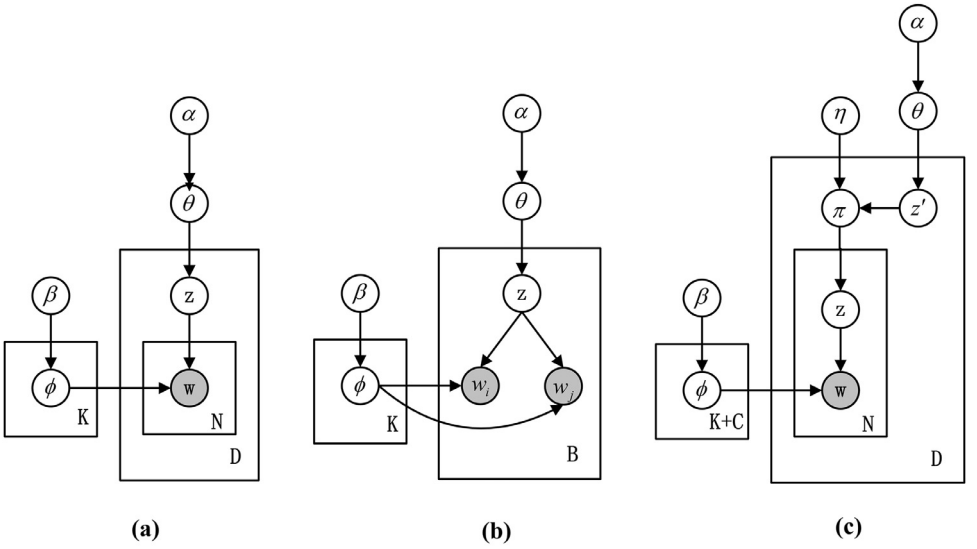

Fig. 1. Graphical model representations of (a) mixture of unigrams, (b) BTM and (c) CSTM.

Another recent discriminative-BTM (d-BTM) model [24] classifies words into three types, including the topical word, general word and document-special word. The general word is similar to the background noise word described in this paper. By removal of general words, d-BTM achieved competitive clustering performance on short texts, however it uses an empirical threshold of document ratio to find general words, which may be biased.

Another mainstream refers to using auxiliary knowledge. An early representative algorithm [19] models short texts by using hidden topics learned from a big reference corpus, e.g., Wikipedia documents. Recently, the works proposed in [1,10,17,22] integrate short text topic modeling with word embeddings. The generalized Pólya urn-Dirichlet multinomial mixture model [10] supposes that similar words (i.e., measured by word embeddings) tend to be clustered together in topics. Following this, it extends the mixture of unigrams by incorporating an additional generalized Pólya urn step into the Gibbs sampling inference process. The distributed representation-based expansion (DREx) method [1] expands short texts by exploiting the word embedding space, and then trains a traditional topic model on the enriched pseudo-documents. Previous studies showed that these algorithms performed well on short text analysis tasks.

Additionally, there are several previous topic models for normal long documents [13,25] using global topics to capture the common semantics. However, these models cannot handle the sparsity problem of short texts, just like LDA. In contrast, CSTM is simple but easy-to-implement for short texts in practice.

2.2. Mixture of unigrams

The mixture of unigrams model [18] assumes that each document concerns a single topic. This assumption can indirectly enrich word token samples for topics at the document level, and then help us modeling the sparse short texts. In the standard mixture of unigrams model, each document d is generated by first choosing a topic z_d from a corpus-level topic distribution θ, and then generating N_d word tokens independently from the topic-word distribution given the selected topic ϕ_{z_d}.

For convenience, in this paper we use the extended mixture of unigrams model that includes Dirichlet priors (i.e., α and β) on the two multinomial distributions (i.e., θ and ϕ). The generative process of this model (Fig. 1a) is given as follows:

1. Sample a corpus-level distribution over topics: $\theta \sim Dirichlet(\alpha)$
2. For each topic k
   a. Sample a topic distribution over words: $\phi_k \sim Dirichlet(\beta)$
3. For each document d
   a. Sample a topic : $z_d \sim Multinomial(\theta)$
   b. For each of the N_d words w_{dn}
      (i.) Sample a word: $w_{dn} \sim Multinomial(\phi_{z_d})$

2.3. BTM

To solve the sparsity problem, BTM [6,26] directly models the rich global word co-occurrence patterns. BTM ignores the document identities and defines an alternative base unit biterm, where any two distinct words in a short text construct a

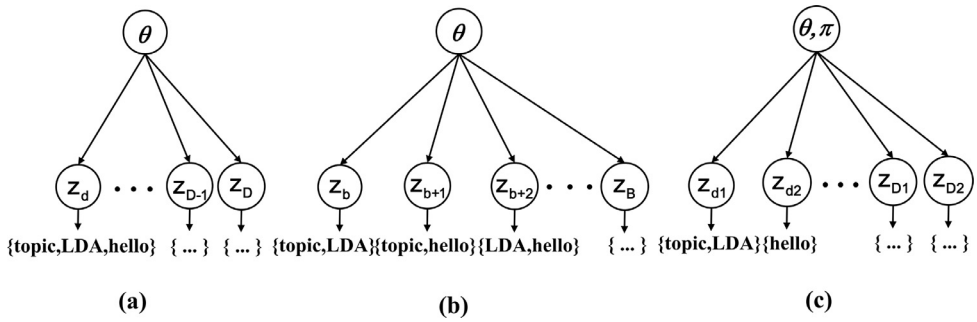

Fig. 2. Suppose there is a short text as {topic, LDA, hello}. Its topic assignments for different topic models: (a) mixture of unigrams, (b) BTM and (c) CSTM.

biterm (i.e., an unordered word co-occurrence pattern). In BTM, the whole corpus is represented by a multinomial distribution θ over topics. Each biterm b is generated by first choosing a topic z_b from the corpus-level topic distribution θ, and then generating both words in this biterm from the selected topic-word distribution ϕ_{z_b}. Supposing that there are total B biterms, the generative process of BTM (Fig. 1b) with Dirichlet priors α and β is given as follows:

1. Sample a distribution over topics: $\theta \sim Dirichlet(\alpha)$
2. For each topic k
   a. Sample a distribution over words: $\phi_k \sim Dirichlet(\beta)$
3. For each of the B biterms b
   a. Sample a topic: $z_b \sim Multinomial(\theta)$
   b. Sample words: $w_{b1}, w_{b2} \sim Multinomial(\phi_{z_b})$

3. The proposed CSTM algorithm

Topic modeling for short texts mainly suffers from two problems, i.e., the sparsity and noise problems. The existing models mainly focus on the sparsity problem, but neglect the noise one. Fig. 2 shows an example of a short text, which contains three words, i.e., {topic, LDA, hello}. Since the word "hello" is frequently occurring but meaningless in most cases, it is regarded as a noise word. Mixture of unigrams (Fig. 2a) models this text by assigning the three words a same topic, and BTM (Fig. 2b) models this text by first constructing three biterms and then assigning each biterm a topic. In both models, the noise word "hello" is forced to be the partner with the informative words "topic" and "LDA". This is obviously problematic.

To further solve the noise problem, we propose an extension of the mixture of unigrams model, namely CSTM. Our CSTM additionally describes a generative process of noise words. This is achieved by introducing a new type of topic, namely common topic, to gather the noise words. In contrast to the common topic, we call the conventional topic function topic. Basically, we assume that each document is represented by a distribution over a mixture of a function topic and all common topics, where the function topic is sampled from a corpus-level topic distribution. The intuition of this setting is that each short text may convey common semantics and only discuss a single special subject. Some previous models also [5,21,28] define the background topic (i.e., the common topic in this paper) to cluster less discriminative words. Reviewing the example short text {topic, LDA, hello}, in CSTM (Fig. 2c) the informative words "topic" and "LDA" can be gathered by a function topic (maybe refer to topic modeling), and the noise word "hello" can be gathered by a common topic.

Let D and V represent the number of documents and unique words; K and C represent the number of function topics and common topics, respectively. The generative process of CSTM is depicted by graphical model notations in Fig. 1c. In step 1, it draws a corpus-level multinomial distribution θ over function topics, from the Dirichlet prior α. In step 2, it draws multinomial distributions ϕ over words for all topics, from the Dirichlet prior β. Step 3 is the word generation process. For each document d, it draws a function topic z'_d from the distribution θ, and then generates a multinomial distribution π_d over the selected function topic z'_d and all common topics, from the Dirichlet prior η. Repeat the following process N_d times for N_d word tokens: it first samples a topic z_{dn} from the distribution π_d, and then samples a word token w_{dn} from the distribution ϕ_{z_d}. Some important notations are listed in Table 1, and the generative process of CSTM is summarized as follows:

1. Sample a distribution over function topics: $\theta \sim Dirichlet(\alpha)$
2. For each of the $K+C$ topics k
   a. Sample a distribution over words: $\phi_k \sim Dirichlet(\beta)$
3. For each document d
   a. Sample a function topic : $z'_d \sim Multinomial(\theta)$
   b. Sample a distribution over a mixture of z'_d and all common topics: $\pi_d \sim Dirichlet(\eta)$
   c. For each of the N_d words w_{dn}
      (i.) Sample a topic: $z_{dn} \sim Multinomial(\pi_d)$

Table 1
Notation descriptions.

Notation	Description
D	The number of documents
V	The number of unique words
K	The number of topics (or function topics in CSTM)
C	The number of common topics in CSTM
(η, π)	The Dirichlet-multinomial pair for document-level topic distributions in CSTM
(α, θ)	The Dirichlet-multinomial pair for the corpus-level topic distributions
(β, ϕ)	The Dirichlet-multinomial pair for topic-word distributions

(ii.) Sample a word: $w_{dn} \sim Multinomial\left(\phi_{z_{dn}}\right)$

Discussion. To further discuss the common topic defined in CSTM, we compare CSTM with two current models, i.e., mixture of unigrams and BTM. All the three models can be expressed from a big document perspective. That is to say, we regard the whole corpus as a big document, which is described by a global distribution over topics, drawn from a Dirichlet prior. In mixture of unigrams, each short text (i.e., a N_d-gram phrase) samples a single topic from the global topic distribution. The word tokens of this N_d-gram phrase are assigned to the same selected topic. BTM breaks the short texts into biterms. Each biterm (i.e., a bi-gram phrase) samples a topic from the global topic distribution. The two words of this biterm are assigned to the same selected topic. Under this expression, it is clear that both models consider the corpus-level information (i.e., the global topic distribution) and the word co-occurrence information about topics (i.e., N_d-gram phrase and bi-gram phrase), contributing to a solution to the sparsity problem. Compared to these two models, our CSTM further considers the noise words. Similar with mixture of unigrams, each short text in CSTM samples a single function topic from the global topic distribution, but each word token is allowed to be sampled from either the selected function topic or a common topic. In other words, CSTM divides a short text into two parts: one is a $|z'_d|$-gram phrase assigned to the selected function topic drawn from the global topic distribution, the other consists of noise words assigned to common topics. In this sense, CSTM alleviates the sparsity and noise problems at the same time.

3.1. Gibbs sampling for CSTM

Given a short text collection W, the inference task of CSTM is to compute the posterior distribution over the latent variables of interest, including the topic-word distribution ϕ, the corpus-level topic distribution θ, the document-level topic distribution π, the function topic assignments for documents z' and the topic assignments for word tokens z. Since it is intractable to compute the exact posterior distribution, we use Gibbs sampling for approximating inference.

Following Griffiths and Steyvers [7], the multinomial distributions ϕ, θ and π can be efficiently marginalized out due to the conjugate Dirichlet-multinomial design. We thus only need to sample the two topic assignments z' and z. We use the Gibbs sampling method, in which z' and z are alternately sampled given all other variables. We directly present the sampling equations, and the derivation details can be found in Appendix.

Similar to the derivation process of mixture of unigrams and LDA, the conditional posterior probability for z' over K function topics is given as follows:

$$p\left(z'_d = k | z'^{-d}, z, W, \alpha, \beta\right) \propto \left(\widehat{N}_k^{-d} + \alpha\right) \frac{\prod_{v=1}^{V} \prod_{n=1}^{N_{dv}} \left(N_{kv}^{-d} + n - 1 + \beta\right)}{\prod_{n=1}^{|z'_d|} \left(N_k^{-d} + n - 1 + V\beta\right)} \tag{1}$$

where $\widehat{N}_k$ is the number of documents assigned to the function topic k; N_{kv} and N_k are the number of word type v and the total number of words assigned to topic k, respectively; N_{dv} is the number of word type v that has occurred in document d; $|z'_d|$ is the number of word tokens assigned to the selected function topic z'_d in document d; the superscript "$-d$" denotes a quantity that excludes the document d.

Additionally, the conditional posterior probability for z over the selected function topic z' and C common topics is given as follows:

$$p\left(z_{dn} = k | z^{-dn}, z', W, \beta, \eta\right) \propto \left(N_{dk}^{-dn} + \eta_k\right) \frac{N_{kw_{dn}}^{-dn} + \beta}{N_k^{-dn} + V\beta} \tag{2}$$

where N_{dk} is the number of word tokens assigned to topic k in document d; the superscript "$-dn$" denotes a quantity that excludes z_{dn} from the position (d, n).

During model inference, the two topic assignments are iteratively sampled by Eqs. (1) and (2) until convergence. The Gibbs sampling inference of CSTM is outlined in Algorithm 1. Finally, we can compute the posterior estimates of ϕ and θ

Algorithm 1 Gibbs sampling for CSTM.

Initialize α, β and η
For $t = 1, 2, \cdots, Max\_iter$ **do**
  For short text d=1 to D **do**
    Sample a function topic assignment z'_d using Eq. (1)
    For word token n=1 to N_d **do**
      Sample a topic assignment z_{dn} using Eq. (2)
    End for
  End for
End for

Table 2
Per-iteration time complexities of Gibbs sampling for LDA, mixture of unigrams (Mix-gram), BTM and CSTM.

Model	Time complexity
LDA	$O(KN_V)$
Mix-gram	$O(KN_V)$
BTM	$O(KB)$
CSTM	$O(CN_V + K\sum_d \lvert z'_d \rvert)$

by:

$$\phi_{kv} = \frac{N_{kv} + \beta}{N_k + V\beta} \tag{3}$$

$$\theta_k = \frac{\widehat{N}_k + \alpha}{D + K\alpha} \tag{4}$$

For π, we use an asymmetric Dirichlet prior η, which assigns the corresponding prior of the function topic a large value η_f, and assigns the corresponding prior of all C common topics a small value η_c. For example, supposing there are two common topics, the Dirichlet prior η actually used is $[\eta_f, \eta_c, \eta_c]$. Using such Dirichlet prior η, we can regulate the proportion of the function topic in a single short text. The posterior estimate of π can be computed by:

$$\pi_{dk} = \frac{N_{dk} + \eta_k}{N_d + \eta_f + C\eta_c} \tag{5}$$

where $\eta_k = \begin{cases} \eta_f & if\ k = z'_d \\ \eta_c & else \end{cases}$

3.2. Time complexity

We discuss the time complexities of Gibbs sampling for LDA, mixture of unigrams, BTM and CSTM. Following Cheng et al. [6], we analyze the time complexity by reviewing the major time consuming part of the Gibbs sampling inference process.

For clarification, again declare that K denotes the number of function topics in CSTM, and denotes the total number of topics in other three models; B denotes the number of biterms in BTM; and N_V denotes the total number of word tokens in a corpus.

For LDA and BTM, the main time cost is the topic assignment sampling of each word token or biterm over K topics, requiring $O(K)$ time. Thus the per-iteration time complexity of LDA is $O(KN_V)$, and that of BTM is $O(KB)$. Unlike the two models, CSTM samples a topic assignment for each word token over C+1 topics, i.e., Eq. (2), which requires $O(C+1)$ time. Additionally, CSTM has to additionally sample a function topic assignment for each document, i.e., Eq. (1), which requires $O(K\lvert z'_d \rvert)$ time. Thus the per-iteration time complexity of CSTM is given by $O(CN_V + K\sum_d \lvert z'_d \rvert)$. Mixture of unigrams can be considered as a special case of CSTM, where the common topic number C is zero, so its per-iteration time complexity is given by $O(K\sum_d N_d)$, which is equivalent to $O(KN_V)$. The time complexities of all models are summarized in Table 2.

Note that the biterm number B is always greater than the word token number N_V, thus BTM must be more time-consuming than other three models. We then compare the time cost of CSTM with those of LDA and mixture of unigrams. The per-iteration time complexity of CSTM can be rearranged as follows:

$$O\left(CN_V + K\sum\nolimits_d \left|z'_d\right|\right) = O\left(\left(\frac{C}{K} + \frac{\sum_d \left|z'_d\right|}{N_V}\right)KN_V\right)$$

Table 3
Summary of the datasets. N_V and *AvgD* denote the total number of word tokens and average document length, respectively. *#Label* denotes the number of categories.

Dataset	D	V	N_V	*AvgD*	*#Label*
Snippets	12,340	30,452	215,367	17.5	8
Question	179,022	26,565	735,793	4.1	35
Tweets	2,000,000	121,788	10,732,356	5.0	–

The comparison now becomes whether the value of the coefficient $\frac{C}{K} + \frac{\sum_d |z'_d|}{N_V}$ is greater than 1. Fortunately, both terms of this coefficient are often small, where the first term is the ratio of the common topics to the function topics, and the second term is the proportion of the function topic assignments in the training corpus. Our early experimental results showed that a few common topics, e.g., C=5 in our experiments, can effectively capture the noise words, thus the first term $\frac{C}{K}$ can be insignificant in practice. Roughly, this coefficient discussed above is always close to 1. That is to say, the time complexity of CSTM is close to those of LDA and mixture of unigrams. Empirical results on run-time can be seen in Section 4.4.4.

4. Experiment

In this section, we evaluate the proposed CSTM algorithm qualitatively and quantitatively.

4.1. *Dataset*

In the experiments, we use three real-world short text datasets, including *Snippets*[1], *Question*[2] and *Tweets*. The *Snippets* dataset was selected from the results of web search transaction using predefined phrases of 8 different categories [19]. The *Question* dataset was collected by Cheng et al. [6]. It was crawled from a popular Chinese Q&A websites, which contains 35 categories. The *Tweets* dataset is a subset of tweets collection published in TREC 2011 microblog track.[3] During pre-processing, the standard stop words[4] and the documents with a single word token were removed. The statistics of datasets are listed in Table 3.

4.2. *Baseline model*

In the experiments, we use six baseline models, including LDA, mixture of unigrams (abbr. Mix-gram), BTM, SATM, Twitter-BTM (abbr. T-BTM) and DREx+LDA [1]. For fair comparisons, all the models are inferred by Gibbs sampling. The iteration number is set to 1000, and the hyper-parameters are set as: $\alpha = 0.1$ and $\beta = 0.01$. Additionally, some model-specific settings are presented as follows:

- **SATM**: SATM supposes that each short text is sampled from a unobserved long pseudo-document. Following Li et al. [11], we tune the pseudo-document number from 100 to 1000 in all evaluations. The best scores are reported finally.
- **T-BTM**: T-BTM is an extension of BTM by aggregating short texts from the same users. Following Chen et al. [5], the Dirichlet prior γ of T-BTM is set 0.5.
- **DREx+LDA**: DREx expands short texts by exploiting the word embedding space and then train the expanded documents using LDA. For English datasets *Tweets* and *Snippets*, we use the pre-trained 100-dimensional GloVe word embeddings[5] trained on tweets; and for the Chinese *Question* dataset, we also use pre-trained embeddings.[6] Following Bicalho et al. [1], the maximum size of expanded documents is set to 60.
- **CSTM**: For the proposed CSTM, the number of common topics is set to 5; η_f and η_c are set to 1 and 0.1, respectively.

4.3. Qualitative evaluation

For qualitative evaluation, we visualize topic assignments for word tokens at the document level and topic distributions learned by different models on the *Tweets* dataset. For baseline models, the number of topics is set to 50; and the function topic number of CSTM is also set to 50.

Additionally, since the highlight of CSTM is noise word filtering, it is necessary to compare it with the traditional pre-processing methods, which also aim to handle the noise words. Toward this end, we use two traditional pre-processing

[1] http://jwebpro.sourceforge.net/data-web-snippets.tar.gz.
[2] http://zhidao.baidu.com.
[3] http://trec.nist.gov/data/tweets/.
[4] CSTM mainly concerns the domain-specific and colloquial noise words, which are difficult to be detected, rather than the standard noise, i.e., stop words. Thus we remove the standard stop words.
[5] https://nlp.stanford.edu/projects/glove/.
[6] https://sites.google.com/site/rmyeid/projects/polyglot.

Table 4
The topic assignments for word tokens of five short texts selected in *Tweets*. The words in bold are assigned to common topics.

Document	Topic assignment
Doc 1	thinking, winner, letter, correct, **rt**
Doc 2	reach, hone, eat, greedy, **lol**
Doc 3	mom, daughter, caralho, **haha**, muito
Doc 4	teen, photo, actions, louder, words
Doc 5	chain, **umcc**, hang, wearing, tonight, push, feds

Table 5
The top 10 word lists of common topics.

Topic	The top 10 words
Common topic 1	rt, don, people, time, lol, good, make, today, ll, twitter
Common topic 2	haha, rt, time, good, live, show, lol, photo, play, great

methods to remove noise words, and then examine the performance of Mix-gram across the pre-filtered datasets. The first method removes high and low document frequency words. The cut-off of high and low frequency words are defined by ρ_h and ρ_l, i.e., a word will be removed if its document frequency is either greater than ρ_h or less than ρ_l. We use the grid search optimization to tune ρ_h and ρ_l on the set $\{2^i|i = 12, 13, 14, 15, 16, 17\}$ and $\{2^i|i = 2, 3, 4, 5, 6, 7\}$, respectively, and then report the best scores.[7] The second method is the textRank algorithm [15], which is initially used to detect keywords. Here, we remove the less important words ranked by textRank. For clarity, we call Mix-gram with the two pre-processing methods Mix-gram$_{Fre}$ and Mix-gram$_{TR}$, respectively.

4.3.1. Evaluation on common topics

First, we evaluate the topic assignments for word tokens inferred by CSTM at the document level. Table 4 presents five short texts from the *Tweets* dataset. We can observe that four words are assigned to common topics, including "rt", "lol", "haha" and "umcc". In terms of these words, three words are frequently occurring background noise words: 1) "rt" is the abbreviation of "retweet". 2) "lol" is the abbreviation of "laugh out loud". 3) "haha" represents a shout of laughter in spoken English. Second, we present the top 10 words of two common topics in Table 5. Besides the words "rt", "lol" and "haha", most of other top words are also background noise words in the *Tweets* dataset. For example, the words "time", "today" and "good" are frequently occurring in *Tweets* posts. In summary, the common topics successfully gather noise words in some degree. This can help CSTM discovering more coherent function topics.

4.3.2. Evaluation on function topics

We evaluate the function topic quality of CSTM by comparing against the topics learned by baseline models. The top 10 words of two *Tweets* topics are presented in Table 6. Overall, we can observe that the topics learned by CSTM are more clean and coherent in most cases. The topics of baseline models include some background noise words, e.g., "rt", "lol" and "time". For the topic about cellphone, LDA and T-BTM include some less relevant words, e.g., "join" and "global"; for the topic about Youtube, Mix-gram includes less relevant words, e.g., "love". Compared with Mix-gram$_{Fre}$ and Mix-gram$_{TR}$, CSTM also performs better. Mix-gram$_{Fre}$ has no background noise words, e.g., "rt" and "lol", however, some other noise words, e.g., "haha" and "mmhmmmm", infiltrate its top word lists. Additionally, we see that Mix-gram$_{TR}$ still contains the background word "lol" and less coherent words. For the topic about cellphone, the word "spell" is less relevant to this topic. Finally, the topics of DREx+LDA seem coherent, but they contain ambiguous top words, e.g., "fruit" and "blackberry" in the topic about cellphone. That is because DREx expands short texts by word embeddings, leading to the enrichment of co-occurrences of similar words, e.g., "apple" and "blackberry". In summary, CSTM qualitatively learns more coherent function topics than baseline models, and performs better than Mix-gram with traditional pre-processing methods. Quantitative evaluations on topic coherence (measured by NPMI) can be found in the following subsection.

4.4. Quantitative evaluation

We quantitatively evaluate CSTM by topic coherence, classification, clustering and run-time.

4.4.1. Topic coherence

A commonly used topic coherence metric is PMI [16], which is based on pointwise mutual information of a reference corpus. To reduce the impact of low frequency counts in word co-occurrences, we employ normalised PMI (NPMI) [4]. Given

[7] The cut-off tuning experiment depends on the performance of NPMI. We introduce this metric in the following subsection.

Table 6
Two top 10 word lists learned by different models. The top section is the topic about cellphone and the bottom section is the topic about Youtube.

Topic	Model	The top 10 words
cellphone	CSTM	iphone, free, app, video, ipad, apple, phone, android, mobile, chat
	LDA	iphone, apple, rt, ipad, news, lol, world, join, ipod, online
	Mix-gram	iphone, free, rt, online, marketing, ipad, app, apple, google, social
	BTM	iphone, apple, rt, ipad, phone, android, app, photo, mobile, camera
	SATM	iphone, lol, apple, world, rt, join, ipod, online, google, chat
	T-BTM	iphone, ipad, lol, global, news, apple, world, join, ipod, online
	DREx+LDA	iphone, apple, fruit, ipad, app, blackberry, microsoft, photo,
	Mix-gram$_{Fre}$	iphone, online, apple, hahahah, world, free, mmhmmmm, google, social, app
	Mix-gram$_{TR}$	iphone, free, lol, spell, ipad, app, apple, photo, social
Youtube	CSTM	youtube, video, channel, subscribed, uploaded, favorited, watch, check, music, official
	LDA	video, lol, youtube, subscribed, good, watch, live, uploaded, channel, favorited
	Mix-gram	video, youtube, love, rt, movie, watch, uploaded, music, favorited, check
	BTM	video, youtube, rt, music, watch, live, uploaded, channel, favorited, subscribed
	SATM	video, youtube, music, live, good, watch, rt, uploaded, channel, film
	T-BTM	video, youtube, check, music, watch, time, live, uploaded, subscribed, videos
	DREx+LDA	video, check, youtube, clips, watch, watching, live, subscribed, uploaded, videos
	Mix-gram$_{Fre}$	video, youtube, haha, watch, movie, huaaaaa, live, music, check, favorited
	Mix-gram$_{TR}$	video, youtube, watch, love, movie, lol, uploaded, music, favorited, live

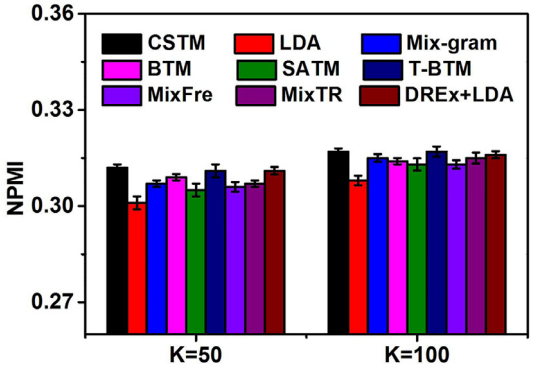

Fig. 3. NPMI results on *Tweets*. MixFre and MixTR represent Mix-grams$_{Fre}$ and Mix-grams$_{TR}$, respectively.

T most probable words in a topic k, its NPMI value is computed by:

$$NPMI(k) = \frac{2}{T(T-1)} \sum_{1 \le i \le j \le T} \frac{\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-\log p(w_i, w_j)} \tag{6}$$

where $p(w_i)$ and $p(w_i, w_j)$ are the probabilities of occurring word w_i and co-occurring word pair (w_i, w_j) estimated by the reference corpus, respectively. In our experiments, an English Wikipedia reference corpus of 8 million documents is used, and T is set to 10.

For all models, we independently run 10 times and report the average NPMI values. For CSTM, only the NPMI values of function topics are computed. The experimental results are shown in Figs. 3 and 4. For baseline models, we can see that the NPMI values of Mix-gram, BTM and SATM are higher than LDA, since short text LDA suffers from severe sparsity problem. In terms of the *Tweets* dataset, T-BTM[8] slightly outperforms BTM. That is because the background topic in T-BTM removes background noise words in some degree. Additionally, we can observe that Mix-gram, Mix-gram$_{Fre}$ and Mix-gram$_{TR}$ are almost at the same level. The performance gap among the three are not obvious. Such results imply that the traditional document pre-processing methods cannot improve the performance of short text topic modeling. Fortunately, we can see that CSTM outperforms baseline models in most cases. The NPMI values of CSTM are higher than those of LDA, Mix-gram, BTM and SATM. More importantly, CSTM is competitive against T-BTM, but it doesn't use any side information such as user ID, which is used in T-BTM.

4.4.2. Document classification

We compare the classification performance among CSTM, LDA, Mix-gram, BTM and SATM on two labelled datasets, i.e., *Snippets* and *Question*. The goal of this experiment is to examine whether CSTM can output more discriminative topic repre-

[8] T-BTM can not be applied to *Snippets* and *Question* datasets, since they contain no user information.

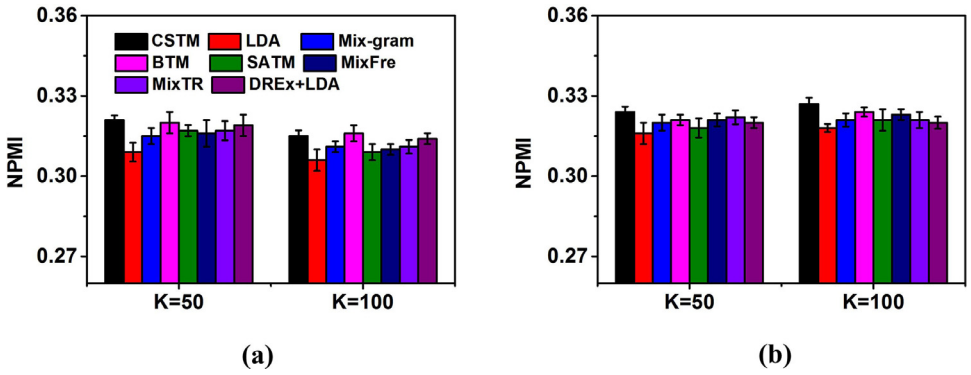

Fig. 4. NPMI results on (a) *Snippets* and (b) *Question*.

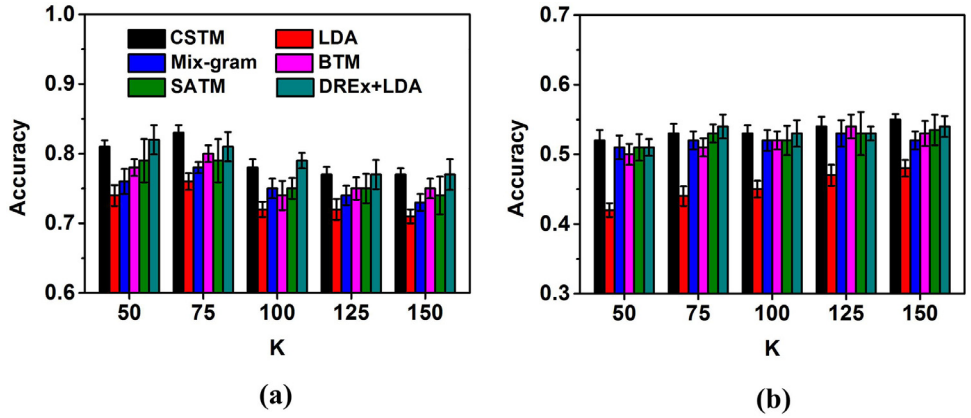

Fig. 5. Classification results on (a) *Snippets* and (b) *Question*.

sentations of documents. To this end, for each model we use the learned topical SW representations [10] as feature vectors, and then feed them into a same classification algorithm. Here, we employ the SVMs implemented by LibSVM.[9] The average classification accuracy scores of 5-fold cross validation are reported.

Fig. 5 presents the classification results with settings of topic number K=50,75,100,125,150. Overall, we can see that CSTM outperforms baseline models in most cases. For example, CSTM performs about 0.02−0.04 better than BTM and about 0.01−0.04 better than SATM. This validates using common topics to filtering out noise words is beneficial to short text topic modeling. Although CSTM performs worse than DREx+LDA in some cases, e.g., the accuracies when K=50, the best scores of CSTM on both datasets are higher than those of DREx+LDA. Additionally, for the *Question* dataset the classification accuracy roughly improves as the topic number increases. But for the *Snippets* dataset the accuracy goes down as the topic number becomes relatively larger, i.e., K=100, 125 and 150. The possible reason is that too many topics are redundant for *Snippets*, which contains only 8 categories.

4.4.3. Document clustering

We evaluate the clustering performance of CSTM on the *Snippets* and *Question* datasets. In the clustering evaluation, each topic corresponds to a cluster. Any document is assigned to the largest topic in its topical SW representation described in [10]. Two evaluation metrics Purity and Entropy are employed. The Purity measures the coherence of a cluster, and it is computed by:

$$Purity = \frac{1}{D}\sum_{k=1}^{K}\max(k) \tag{7}$$

where $\max(k)$ denotes the document number of the dominant category in cluster k. The Entropy measures how homogeneous a cluster is, and it is computed by:

$$Entropy = -\sum_{k=1}^{K}\frac{C_k}{D}\sum_{l=1}^{L}\frac{C_{kl}}{C_k}\log\left(\frac{C_{kl}}{C_k}\right) \tag{8}$$

[9] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

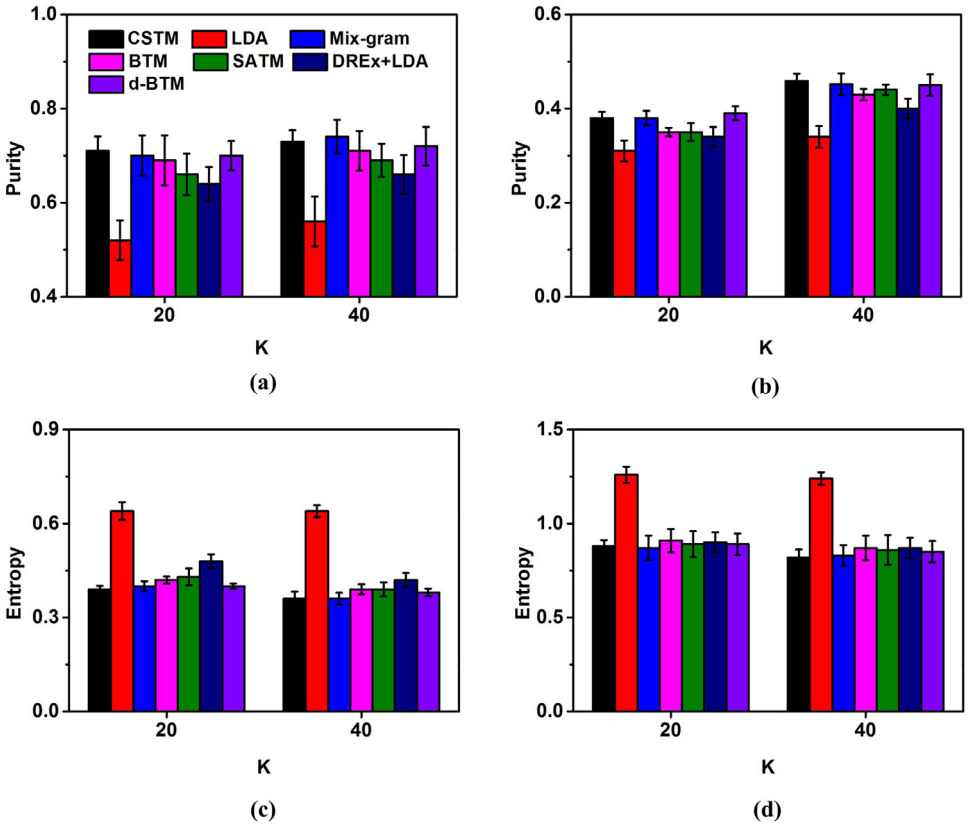

Fig. 6. Clustering results of Purity on (a) *Snippets* and (b) *Question*, and Entropy on (c) *Snippets* and (d) *Question*. Higher/lower values of Purity/Entropy imply better performance.

where L is the number of categories; C_{kl} and C_k are the number of documents labeled by category l in cluster k and the total number of documents in cluster k, respectively. Note that higher Purity and lower Entropy scores imply better clustering performance.

Besides the baseline models used in classification experiments, we also compare CSTM against the d-BTM(-G) [24] model, which empirically removes general words. Fig. 6 shows the clustering results with different cluster numbers, i.e., $K = 20, 40$. First, we can see that CSTM outperforms LDA, Mix-gram, BTM, SATM and DREx+LDA. This indicates that using common topics to capture background noise words can improve the clustering performance. Second, CSTM performs slightly better than d-BTM(-G), which also filters out general words. The possible reason is that d-BTM(-G) removes general words using an empirical cutoff, resulting in the loss of useful words in some degree.

4.4.4. Run-time

We empirically compare the run-time among LDA, Mix-gram, BTM and CSTM. For each model, we run a Gibbs sampling process of 10,000 iterations and compute the average per-iteration run-time. As shown in Fig. 7, it can be seen that the efficiency gap among LDA, Mix-gram and CSTM is not obvious. BTM runs slower than other three models, especially for the *Snippets* dataset. That is because the average document length of *Snippets* is relatively larger (i.e., 17.5), resulting in too many biterms. In summary, the empirical run-time results validate the analysis of time complexity in Section 3.2.

4.5. Evaluation of the number of common topics

We are regardful of whether CSTM is sensitive to the number of common topics C. To answer this question, we train a CSTM model with 50 function topics, and then evaluate the performance of NPMI with different C values from 1 to 10. The average results of 10 runs are reported. As shown in Fig. 8, we can observe that CSTM is not very sensitive to the common topic number. The best NPMI scores are achieved when C=4 and 5 on all the three datasets. In previous experiments, we have also evaluated the performance of clustering and classification with difference C values. Those results are very similar to the NPMI results. The performance is always better when C=4 and 5. In practice, the suggestion common topic number C is set to 5.

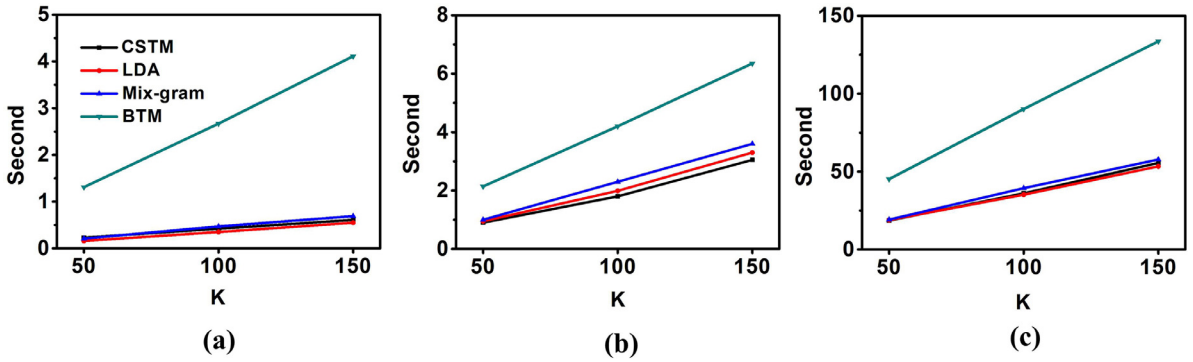

Fig. 7. Per-iteration time cost on (a) *Snippets*, (b) *Question* and (c) *Tweets*.

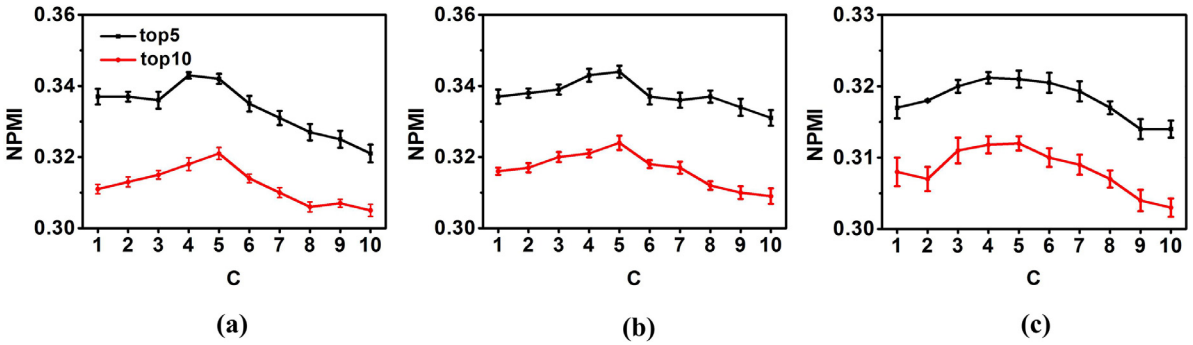

Fig. 8. NPMI results with different C values on (a) *Snippets*, (b) *Question* and (c) *Tweets*.

5. Conclusion

In this paper, we investigate a generalized topic model CSTM for short texts. The key idea is to capture the background noise words by introducing a new type of topic, namely common topic. The experimental results on three real-world datasets indicate that CSTM can learn more coherent topics, and it is competitive against the state-of-the-art short text topic models on classification and clustering tasks. Furthermore, the run-time of CSTM is at the same level of LDA and Mix-gram models.

There are some limitations of CSTM that we haven't fully discussed and evaluated in this work. The first is how to set the number of common topics C for different datasets. The second refers to the settings of priors η_f and η_c. These two hyper-parameters reflect our prior on the proportion of function topics, which may be also sensitive to different datasets. In the future, we plan to further investigate these two problems of CSTM in practice.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (NSFC) [61602204 and 61472157].

Appendix A

We now derive the Gibbs sampling equations of function topic assignments for documents z' and the topic assignments for word tokens z. Toward this goal, we first present the joint distribution of z' and z given a corpus W:

$$
\begin{aligned}
p\left(W, z', z|\alpha, \beta, \eta\right) &= \int\int\int p\left(W, \theta, \phi, z', z|\alpha, \beta, \eta\right) d\theta d\phi d\pi \\
&= \int\int\int \mathrm{Dir}(\theta|\alpha) \prod_{k=1}^{K} \theta_k^{\widehat{N}_k} \prod_{k=1}^{K+C} \mathrm{Dir}(\phi_k|\beta) \prod_{v=1}^{V} \phi_{kv}^{N_{kv}} \prod_{d=1}^{D} \mathrm{Dir}(\pi_d|\eta) \prod_{k\in\Omega_d} \pi_{dk}^{N_{dk}} d\theta d\phi d\pi \\
&= \left(\frac{\prod_{k=1}^{K} \Gamma\left(\widehat{N}_k + \alpha\right)}{\Gamma(D + K\alpha)} \frac{\Gamma(K\alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)} \right) \left(\prod_{k=1}^{K+C} \frac{\prod_{v=1}^{V} \Gamma(N_{kv} + \beta)}{\Gamma(N_k + V\beta)} \frac{\Gamma(V\beta)}{\prod_{v=1}^{V} \Gamma(\beta)} \right)
\end{aligned}
$$

$$\times\left(\prod_{d=1}^{D}\frac{\prod_{k\in\Omega_d}\Gamma(N_{dk}+\eta_k)}{\Gamma\left(N_d+\eta_f+C\eta_c\right)}\frac{\Gamma\left(\eta_f+C\eta_c\right)}{\prod_{k\in\Omega_d}\Gamma(\eta_k)}\right)$$
$$\propto\left(\frac{\prod_{k=1}^{K}\Gamma\left(\widehat{N}_k+\alpha\right)}{\Gamma(D+K\alpha)}\right)\left(\prod_{k=1}^{K+C}\frac{\prod_{v=1}^{V}\Gamma(N_{kv}+\beta)}{\Gamma(N_k+V\beta)}\right)\left(\prod_{d=1}^{D}\frac{\prod_{k\in\Omega_d}\Gamma(N_{dk}+\eta_k)}{\Gamma\left(N_d+\eta_f+C\eta_c\right)}\right)$$
$$\triangleq B\left(\widehat{N}_k,D,\alpha\right)B\left(N_{kv},N_{k,}\beta\right)B\left(N_{dk},N_{d,}\eta\right) \tag{9}$$

where Ω_d is the mixture of the selected function topic z'_d and all C common topics. Besides, we use the notation $B(\cdot)$ to represent the corresponding terms in the fifth line for convenience. For example, $B\left(\widehat{N}_k,D,\alpha\right)$ describes the term $\frac{\prod_{k=1}^{K}\Gamma\left(\widehat{N}_k+\alpha\right)}{\Gamma(D+K\alpha)}$.

The proposed CSTM alternately draws the samples for z' and z by holding the other one fixed. We now show the derivations of sampling equations.

Sampling equation of z'given the current z

In the context of Gibbs sampling, we draw a function topic assignment z'_d from the posterior distribution conditioned on all other variables:

$$p\left(z'_d|z'^{-d},z,W,\alpha,\beta,\eta\right)=\frac{p\left(W,z'|z,\alpha,\beta,\eta\right)}{p\left(W,z'^{-d}|z,\alpha,\beta,\eta\right)}\propto\frac{p\left(W,z'|z,\alpha,\beta,\eta\right)}{p\left(W^{-d},z'^{-d}|z,\alpha,\beta,\eta\right)}$$
$$=\frac{p\left(W,z',z|\alpha,\beta,\eta\right)}{p\left(W^{-d},z'^{-d},z|\alpha,\beta,\eta\right)}\propto\frac{p\left(W,z',z|\alpha,\beta,\eta\right)}{p\left(W^{-d},z'^{-d},z^{-d}|\alpha,\beta,\eta\right)} \tag{10}$$

Putting the joint distribution of Eq. (9) into Eq. (10), we have:

$$p\left(z'_d|z'^{-d},z,W,\alpha,\beta,\eta\right)\propto\frac{B\left(\widehat{N}_k,D,\alpha\right)B\left(N_{kv},N_{k,}\beta\right)B\left(N_{dk},N_{d,}\eta\right)}{B\left(\widehat{N}_k^{-d},D-1,\alpha\right)B\left(N_{kv}^{-d},N_k^{-d},\beta\right)B\left(N_{dk}^{-d},N_d^{-d},\eta\right)}$$
$$\propto\frac{B\left(\widehat{N}_k,D,\alpha\right)B\left(N_{kv},N_{k,}\beta\right)}{B\left(\widehat{N}_k^{-d},D-1,\alpha\right)B\left(N_{kv}^{-d},N_k^{-d},\beta\right)} \tag{11}$$

The second line in Eq. (11) follows that the counts of document-level topic assignments are independent of the function topic assignment z'_d. Expanding Eq. (11), we can obtain the Gibbs sampling equation of z':

$$p\left(z'_d=k|z'^{-d},z,W,\alpha,\beta\right)\propto\frac{\left(\frac{\prod_{k=1}^{K}\Gamma\left(\widehat{N}_k+\alpha\right)}{\Gamma(D+K\alpha)}\right)\left(\prod_{k=1}^{K+C}\frac{\prod_{v=1}^{V}\Gamma(N_{kv}+\beta)}{\Gamma(N_k+V\beta)}\right)}{\left(\frac{\prod_{k=1}^{K}\Gamma\left(\widehat{N}_k^{-d}+\alpha\right)}{\Gamma(D-1+K\alpha)}\right)\left(\prod_{k=1}^{K+C}\frac{\prod_{v=1}^{V}\Gamma\left(N_{kv}^{-d}+\beta\right)}{\Gamma\left(N_k^{-d}+V\beta\right)}\right)}$$
$$=\frac{\Gamma\left(\widehat{N}_k+\alpha\right)}{\Gamma\left(\widehat{N}_k^{-d}+\alpha\right)}\frac{\Gamma(D-1+K\alpha)}{\Gamma(D+K\alpha)}\frac{\prod_{v=1}^{V}\Gamma(N_{kv}+\beta)}{\prod_{v=1}^{V}\Gamma\left(N_{kv}^{-d}+\beta\right)}\frac{\Gamma\left(N_k^{-d}+V\beta\right)}{\Gamma(N_k+V\beta)}$$
$$\propto\left(\widehat{N}_k^{-d}+\alpha\right)\frac{\prod_{v=1}^{V}\prod_{n=1}^{N_{dv}}\left(N_{kv}^{-d}+n-1+\beta\right)}{\prod_{n=1}^{|z'_d|}\left(N_k^{-d}+n-1+V\beta\right)} \tag{12}$$

Sampling equation of z given the current z'

The derivation of z sampling is similar with that of z'. The posterior distribution of z_{dn} conditioned on all other variables is as follows:

$$p\left(z_{dn}|z^{-dn},z',W,\alpha,\beta,\eta\right)=\frac{p\left(W,z|z',\alpha,\beta,\eta\right)}{p\left(W,z^{-dn}|z',\alpha,\beta,\eta\right)}\propto\frac{p\left(W,z|z',\alpha,\beta,\eta\right)}{p\left(W^{-dn},z^{-dn}|z',\alpha,\beta,\eta\right)}$$
$$=\frac{p\left(W,z,z'|\alpha,\beta,\eta\right)}{p\left(W^{-dn},z^{-dn},z'|\alpha,\beta,\eta\right)}\propto\frac{p\left(W,z',z|\alpha,\beta,\eta\right)}{p\left(W^{-dn},z^{-dn},z'^{-dn}|\alpha,\beta,\eta\right)}$$
$$=\frac{B\left(\widehat{N}_k,D,\alpha\right)B(N_{kv},N_k,\beta)B(N_{dk},N_d,\eta)}{B\left(\widehat{N}_k^{-dn},D,\alpha\right)B\left(N_{kv}^{-dn},N_k^{-dn},\beta\right)B\left(N_{dk}^{-dn},N_d^{-dn},\eta\right)}$$
$$\propto\frac{B(N_{kv},N_k,\beta)B(N_{dk},N_d,\eta)}{B\left(N_{kv}^{-dn},N_k^{-dn},\beta\right)B\left(N_{dk}^{-dn},N_d^{-dn},\eta\right)} \tag{13}$$

Expanding Eq. (13), we can obtain the Gibbs sampling equation for z:

$$
\begin{aligned}
p\left(z_{dn}=k|z^{-dn},z',W,\beta,\eta\right) &\propto \frac{\left(\prod_{k=1}^{K+C}\frac{\prod_{v=1}^{V}\Gamma\left(N_{kv}+\beta\right)}{\Gamma\left(N_k+V\beta\right)}\right)\left(\prod_{d=1}^{D}\frac{\prod_{k\in\Omega_d}\Gamma\left(N_{dk}+\eta_k\right)}{\Gamma\left(N_d+\eta_f+C\eta_c\right)}\right)}{\left(\prod_{k=1}^{K+C}\frac{\prod_{v=1}^{V}\Gamma\left(N_{kv}^{-dn}+\beta\right)}{\Gamma\left(N_k^{-dn}+V\beta\right)}\right)\left(\prod_{d=1}^{D}\frac{\prod_{k\in\Omega_d}\Gamma\left(N_{dk}^{-dn}+\eta_k\right)}{\Gamma\left(N_d^{-dn}+\eta_f+C\eta_c\right)}\right)} \\
&= \frac{\Gamma\left(N_{kw_{dn}}+\beta\right)}{\Gamma\left(N_{kw_{dn}}^{-dn}+\beta\right)}\frac{\Gamma\left(N_k^{-dn}+V\beta\right)}{\Gamma\left(N_k+V\beta\right)}\frac{\Gamma\left(N_{dk}+\eta_k\right)}{\Gamma\left(N_{dk}^{-dn}+\eta_k\right)}\frac{\Gamma\left(N_d^{-dn}+\eta_f+C\eta_c\right)}{\Gamma\left(N_d+\eta_f+C\eta_c\right)} \\
&\propto \left(N_{dk}^{-dn}+\eta_k\right)\frac{N_{kw_{dn}}^{-dn}+\beta}{N_k^{-dn}+V\beta}
\end{aligned}
\tag{14}
$$

References

[1] P. Bicalho, M. Pita, G. Pedrosa, A. Lacerda, G.L. Pappa, A general framework to expand short text for topic modeling, Inf. Sci. 393 (2017) 66–81.
[2] D.M. Blei, J.D. Lafferty, A correlated topic model fo science, Ann. Appl. Stat. 1 (1) (2007) 17–35.
[3] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[4] G. Bouma, Normalized (pointwise) mutual information in collocation extraction, in: Proceedings of the Biennial GSCL Conference, 2009, pp. 31–40.
[5] W. Chen, J. Wang, Y. Zhang, H. Yan, X. Li, User based aggregation for biterm topic model, in: Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, 2015, pp. 489–494.
[6] X. Cheng, X. Yan, Y. Lan, J. Guo, BTM: Topic modeling over short texts, IEEE Trans. Knowl. Data Eng. 26 (12) (2014) 2928–2941.
[7] T.L. Griffiths, M. Steyvers, Finding scientific topics, Natl. Acad. Sci. United States Am. 101 (Suppl. 1) (2004) 5228–5235.
[8] L. Hong, B.D. Davison, Empirical study of topic modeling in Twitter, in: In Proceedings of the First Workshop on Social Media Analytics, 2010, pp. 80–88.
[9] H. Lakkaraju, I. Bhattacharya, C. Bhattacharyys, Dynamic multi-relational Chinese restaurant process for analyzing influences on users in social media, in: IEEE International Conference on Data Mining, 2012, pp. 389–398.
[10] C. Li, H. Wang, Z. Zhang, A. Sun, Z. Ma, Topic modeling for short texts with auxiliary word embeddingsw, in: International ACM SIGIR conference on Research and Development in Information Retrieval, 2016, pp. 165–174.
[11] C. Li, H. Wang, Z. Zhang, X. Sun, Z. Ma, Topic modeling for short texts with auxiliary word embeddings, in: International Conference on Research and Development in Information Retrieval, 2016, pp. 165–174.
[12] X. Li, C. Li, J. Chi, J. Ouyang, Short text topic modeling by exploring original documentss, Knowl. Inf. Syst. (2017), doi:10.1007/s10115-017-1099-0.
[13] X. Li, J. Ouyang, Y. Lu, X. Zhou, T. Tian, Group topic model: organizing topics into groups, Inf. Retr. 18 (1) (2015) 1–25.
[14] R. Mehrotra, S. Sanner, W. Buntine, L. Xie, Improving lda topic models for microblogs via Tweet pooling and automatic labeling, in: International ACM SIGIR Conference on Research and Development in Information Retrieval, 2013, pp. 889–892.
[15] R. Mihalcea, P. Tarau, Textrank: Bringing order into texts, in: Conference on Empirical Methods in Natural Language Processing, 2004, pp. 404–411.
[16] D. Newman, J.H. Lau, K. Grieser, T. Baldwin, Automatic evaluation of topic coherence, in: Annual Conference of the North American Chapter of the ACL, 2010, pp. 100–108.
[17] D.Q. Nguyen, R. Billingsley, L. Du, M. Johnson, Improving topic models with latent feature word representations, Trans. Assoc. Comput. Linguist. 3 (2015) 299–313.
[18] K. Nigam, A.K. Mccallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using em, Mach. Learn. 39 (2) (2000) 103–134.
[19] X.-H. Phan, L.-M. Nguyen, S. Horiguchi, Learning to classify short and sparse text & web with hidden topics from large-scale data collections, in: International Conference on World Wide Web, 2008, pp. 91–100.
[20] X. Quan, C. Kit, Y. Ge, S.J. Pan, Short and sparse text topic modeling via self-aggregation, in: International Joint Conference on Artificial Intelligenc, 2015, pp. 2270–2276.
[21] K. Sasaki, T. Yoshikawa, T. Furuhashi, Twitter-TTM: An efficient online topic modeling for twitter considering dynamics of user interests and topic trends, in: International Symposium on Soft Computing and Intelligent Systems, Joint 7th International Conference on and Advanced Intelligent Systems, 2014, pp. 440–445.
[22] V.K.R. Sridhar, Unsupervised topic modeling for short texts using distributed representations of words, in: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015. 1192–200
[23] J. Weng, E.-P. Lim, J. Jiang, Q. He, Twitterrank: Finding topic-sensitive influential Twitterers, in: ACM International Conference on Web Search and Data Mining, 2010, pp. 261–270.
[24] Y. Xia, N. Tang, A. Hussain, E. Cambria, Discriminative bi-term topic model for headline-based social news clustering, in: International Florida Artificial Intelligence Research Society Conference, 2015, pp. 311–316.
[25] P. Xie, E.P. Xing, Integrating document clustering and topic modeling, in: Conference on Uncertainty in Artificial Intelligence, 2013, pp. 694–703.
[26] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: International Conference on World Wide Web, 2013, pp. 1445–1456.
[27] J. Yin, J. Wang, A Dirichlet multinomial mixture model-based approach for short text clustering, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 233–242.
[28] W.X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, X. Li, Comparing Twitter and traditional media using topic models, in: European Conference on Advances in Information Retrieval, 2011, pp. 338–349.
[29] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, H. Xiong, Topic modeling of short texts: A pseudo-document view, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 2105–2114.
[30] Y. Zuo, J. Zhao, K. Xu, Word network topic model: a simple but general solution for short and imbalanced texts, Knowl. Inf. Syst. 48 (2) (2016) 379–398.