# Short Text Topic Modeling Techniques, Applications, and Performance: A Survey

Jipeng Qiang, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu, *Fellow, IEEE*

**Abstract**—Analyzing short texts infers discriminative and coherent latent topics that is a critical and fundamental task since many real-world applications require semantic understanding of short texts. Traditional long text topic modeling algorithms (e.g., PLSA and LDA) based on word co-occurrences cannot solve this problem very well since only very limited word co-occurrence information is available in short texts. Therefore, short text topic modeling has already attracted much attention from the machine learning research community in recent years, which aims at overcoming the problem of sparseness in short texts. In this survey, we conduct a comprehensive review of various short text topic modeling techniques proposed in the literature. We present three categories of methods based on Dirichlet multinomial mixture, global word co-occurrences, and self-aggregation, with example of representative approaches in each category and analysis of their performance on various tasks. We develop the first comprehensive open-source library, called STTM, for use in Java that integrates all surveyed algorithms within a unified interface, benchmark datasets, to facilitate the expansion of new methods in this research field. Finally, we evaluate these state-of-the-art methods on many real-world datasets and compare their performance against one another and versus long text topic modeling algorithm.

**Index Terms**—Topic modeling, short text, sparseness, short text topic modeling

## 1 Introduction

Short texts have become an important information source including news headlines, status updates, web page snippets, tweets, question/answer pairs, etc. Short text analysis has been attracting increasing attention in recent years due to the ubiquity of short text in the real-world [1], [2], [3]. Effective and efficient models infer the latent topics from short texts, which can help discover the latent semantic structures that occur in a collection of documents. Short text topic modeling algorithms are always applied into many tasks such as topic detection [4], classification [5], comment summarization [6], user interest profiling [7].

Traditional topic modeling algorithms such as probabilistic latent semantic analysis (PLSA) [8] and latent Dirichlet allocation (LDA) [9] are widely adopted for discovering latent semantic structure from text corpus without requiring any prior annotations or labeling of the documents. In these algorithms, each document may be viewed as a mixture of various topics and each topic is characterized by a distribution over all the words. Statistical techniques (e.g., Variational methods and Gibbs sampling) are then employed to infer the latent topic distribution of each document and the word distribution of each topic using higher-order word co-occurrence patterns [10]. These algorithms and their variants have had a major impact on numerous applied fields in modeling text collections news articles, research papers, and blogs [11], [12], [13]. However, traditional topic models experience large performance degradation over short texts due to the lack of word co-occurrence information in each short text [1], [14]. Therefore, short text topic modeling has already attracted much attention from the machine learning research community in recent years, which aims at overcoming the problem of sparseness in short texts.

Earlier works [15], [16] still used traditional topic models for short texts, but exploited external knowledge or metadata to bring in additional useful word co-occurrences across short texts, and therefore may boost the performance of topic models. For example, Phan *et al.* [16] first learned latent topics from Wikipedia, and then inferred topics from short texts. Weng *et al.* [7] and Mehrotra *et al.* [17] aggregated tweets for pseudo-document using hashtags and the same user respectively. Wang *et al.* [18] first constructed different kinds of hashtag graphs based on hashtags, and proposed a novel framework of hashtag graph-based topic modeling to learn topics. The problem lies in that auxiliary information or metadata is not always available or just too costly for deployment. These studies suggest that topic models specifically designed for general short texts are imperative. This survey will provide a taxonomy that captures the existing short text topic modeling algorithms and their application domains.

News aggregation websites often rely on news headlines to cluster different source news about the same event. In Table 1, we show an event about artificial intelligence reported on March 1, 2018. As presented, all these short texts were reported about the same event. From these short texts, we can found these following characteristics:

- *J. Qiang, Z. Qian, Y. Li, and Y. Yuan are with the Department of Computer Science, Yangzhou University, Jiangsu 225009, China. E-mail: {jpqiang, liyun, yhyuan}@yzu.edu.cn, 289299522@qq.com.*
- *X. Wu is with the Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, Hefei University of Technology, Hefei, Anhui 230026, China, and also with the Mininglamp Academy of Sciences, Mininglamp, Beijing, China. E-mail: xwu@hfut.edu.cn.*

*Manuscript received 16 Apr. 2019; revised 23 Mar. 2020; accepted 30 Apr. 2020. Date of publication 4 May 2020; date of current version 3 Feb. 2022.*
*(Corresponding author: Jipeng Qiang.)*
*Recommended for acceptance by L. Chen.*
*Digital Object Identifier no. 10.1109/TKDE.2020.2992485*

TABLE 1
An Event About Artificial Intelligence was Reported by Several News Media on March 1, 2018

| Number | Media | Headline |
|---|---|---|
| 1 | Lawfare | President Trump's Executive Order on Artificial Intelligence |
| 2 | Nextgov | White Houses Race to Maintain AI Dominance Misses Opportunity |
| 3 | Forbes | Artificial Intelligence Regulation may be Impossible |
| 4 | CognitiveWorld | Pop Culture, AI and Ethics |

1) Each short text lacks enough word co-occurrence information.
2) Due to a few words in each text, most texts are probably generated by only one topic (e.g, text 1, text 2, text 3).
3) Statistical information of words among texts cannot fully capture words that are semantically related but rarely co-occur. For example, President Trump of text 1 and White House of text 2 are highly semantically related, and AI is short for Artificial Intelligence.
4) The single-topic assumption may be too strong for some short texts. For example, text 3 is probably associated with a small number of topics (e.g., one to three topics).

Considering these characteristics, existing short text topic modeling algorithms were proposed by trying to solve one or two of these characteristics. According to the adopted strategies for solving the sparseness problem, we divide the existing work into the three major categories: the first one is based on the assumption that each document is inferred from only one topic; the second one is based on the assumption that two words in one sliding window from one document are sampled from the same topic; the third one is based on the idea that long pseudo-document contains enough word co-occurrence information, so short texts should be merged into long pseudo-documents before topic inference. The following three major categories will be discussed in detail below.

*1) Dirichlet Multinomial Mixture (DMM) Based Methods*. A simple and effective model, Dirichlet Multinomial Mixture model, has been adopted to infer latent topics in short texts [19], [20]. DMM follows the simple assumption that each text is sampled from only one latent topic. Considering the characteristics (1) and (2) in short texts, this assumption is reasonable and suitable for short texts compared to the complex assumption adopted by LDA that each text is modeled over a set of topics [21], [22]. Nigam *et al.* [23] proposed an EM-based algorithm for Dirichlet Multinomial Mixture (DMM) model. Except for the basic expectation maximization (EM), several inference methods have been used to estimate the parameters including variation inference and Gibbs sampling. For example, Yu *et al.* [24] proposed the DMAFP model based on variational inference algorithm [25]. Yin *et al.* [19] proposed a collapsed Gibbs sampling algorithm for DMM. Other variations based on DMM [26], [27], [28] were proposed for improving the performance. The above models based on DMM ignore the characteristic (3). Therefore, many models by incorporating word embeddings into DMM were proposed [29], [30], because word embeddings learned from millions of external documents contain semantic information of words [31]. Not only word co-occurrence words belong to one topic, but words with high similarity have a high probability belonging to one topic, which can effectively solve the data sparsity issue. To highlight the characteristic (4), a Poisson-based DMM model (PDMM) was proposed that allows each short text is sampled by a limited number of topics [32]. Accordingly, Li *et al.* [32] proposed a new model by directly extending the PDMM model using word embeddings.

*2) Global Word Co-Occurrences Based Methods*. Considering the characteristic (1), some models try to use the rich global word co-occurrence patterns for inferring latent topics [14], [33]. Due to the adequacy of global word co-occurrences, the sparsity of short texts is mitigated for these models. According to the utilizing strategies of global word co-occurrences, this type of models can be divided into two types. 1) The first type directly uses the global word co-occurrences to infer latent topics. Biterm topic modeling (BTM) [14] posits that the two words in a biterm share the same topic drawn from a mixture of topics over the whole corpus. Some models extend the Biterm Topic Modeling (BTM) by incorporating the burstiness of biterms as prior knowledge [22] or distinguishing background words from topical words [34]. 2) The second type first constructs word co-occurrence network using global word co-occurrences and then infers latent topics from this network, where each word corresponds to one node and the weight of each edge stands for the empirical co-occurrence probability of the connected two words [33], [35].

*3) Self-Aggregation Based Methods*. Self-aggregation based methods are proposed to perform topic modeling and text self-aggregation during topic inference simultaneously. Short texts are merged into long pseudo-documents before topic inference that can help improve word co-occurrence information. Different from the aforementioned aggregation strategies [7], [17], this type of method SATM [21] and PTM[36] posit that each short text is sampled from a long pseudo-document unobserved in current text collection, and infer latent topics from long pseudo-documents, without depending on auxiliary information or metadata. Considering the characteristic (3), Qiang *et al.* [37] and Bicalho *et al.* [38] merged short texts into long pseudo-documents using word embeddings.

### 1.1 Our Contributions

This survey has the following three-pronged contribution:

1) We propose a taxonomy of algorithms for short text topic modeling and explain their differences. We define three different tasks, i.e., application domains of short text topic modeling techniques. We illustrate the evolution of the topic, the challenges it faces, and future possible research directions.

2) To facilitate the expansion of new methods in this field, we develop the first comprehensive open-source JAVA library, called STTM, which not only includes all short text topic modeling algorithms discussed in this survey with a uniform easy-to-use programming interface but also includes a great number of designed modules for the evaluation and application of short text topic modeling algorithms. STTM is open-sourced at https://github.com/qiang2100/STTM.
3) We finally provide a detailed analysis of short text topic modeling techniques and discuss their performance on various applications. For each method, we analyze their results through a comprehensive comparative evaluation of six common datasets.

### 1.2 Organization of the Survey

The rest of this survey is organized as follows. In Section 2, we introduce the task of short text topic modeling. Section 3 proposes a taxonomy of short text topic modeling algorithms and describes representative approaches in each category. The list of applications for which researchers have used the short text topic modeling algorithms is provided in Section 4. Section 5 presents our Java library for short text topic modeling algorithms. In the next two sections, we describe the experimental setup (Section 6) and evaluate the discussed models (Section 7). Finally, we draw our conclusions and discuss potential future research directions in Section 8.

## 2 Short Text Topic Modeling

In this section, we formally define the problem of short text topic modeling.

Given a short text corpus $\boldsymbol{D}$ of $N$ documents, with a vocabulary $W$ of size $V$, and $K$ pre-defined latent topics. One document $d$ is represented as $(w_{d,1}, w_{d,2}, \ldots, w_{d,n_d})$ in $D$ including $n_d$ words.

A topic $\phi$ in a given collection $\boldsymbol{D}$ is defined as a multinomial distribution over the vocabulary $W$, i.e., $\{p(w|\phi)\}_{w \in W}$. The topic representation of a document $d$, $\theta_d$, is defined as a multinomial distribution over $K$ topics, i.e., $\{p(\phi_k|\theta_d)\}_{k=1,\ldots,K}$. The general task of topic modeling aims to find $K$ salient topics $\phi_{k=1,\ldots,K}$ from $\boldsymbol{D}$ and to find the topic representation of each document $\theta_{d=1,\ldots,N}$.

Most classical probabilistic topic models adopt the Dirichlet prior for both the topics and the topic representation of documents, which are first used in LDA [9], which is $\phi_k \sim Dirichlet(\beta)$ and $\theta_d \sim Dirichlet(\alpha)$. In practice, the Dirichlet prior smooths the topic mixture in individual documents and the word distribution of each topic, which alleviates the overfitting problem of probabilistic latent semantic analysis (PLSA) [8], especially when the number of topics and the size of vocabulary increase. Therefore, all of existing short text topic modeling algorithms adopt Dirichlet distribution as prior distribution.

Given a short text corpus $\boldsymbol{D}$ with a vocabulary of size $V$, and the predefined number of topics $K$, the major tasks of short text topic modeling can be defined as to:

1) Learn the word representation of topics $\phi$;
2) Learn the sparse topic representation of documents $\theta$.

TABLE 2
The Notations of Symbols Used in the Paper

| | |
|---|---|
| $\boldsymbol{D}, N$ | Documents and number of documents in the corpus |
| $W, V$ | The vocabulary and number of words in the vocabulary |
| $K$ | Number of pre-defined latent topics |
| $\bar{l}$ | Average length of each document in $\boldsymbol{D}$ |
| $n_k$ | Number of words associated with topic $k$ |
| $m_k$ | Number of documents associated with topic $k$ |
| $n_k^w$ | Number of word $w$ associated with topic $k$ in $\overrightarrow{d}$ |
| $n_d$ | Number of words in document $d$ |
| $n_d^w$ | Number of word $w$ in document $d$ |
| $n_d^k$ | Number of word associated with topic $k$ in document $d$ |
| $n_{k,d}^w$ | Number of word $w$ associated with topic $k$ in document $d$ |
| $P$ | Long pseudo-document set generated by models |
| $\phi$ | Topic distribution |
| $\theta$ | Document-topic distribution |
| $z$ | Topic indicator |
| $U$ | Number of dimensions in word embeddings |
| $\zeta$ | Time cost of considering GPU model |
| $\varsigma$ | Maximum number of topics allowable in a short text |
| $c$ | Size of sliding window |

All the notations used in this paper are summarized in Table 2.

## 3 Algorithmic Approaches: A Taxonomy

In the past decade, there has been much work to discover latent topics from short texts using traditional topic modeling algorithms by incorporating external knowledge or metadata. More recently, researchers focused on proposing new short text topic modeling algorithms. In the following, we present historical context about the research progress in this domain, then propose a taxonomy of short text topic modeling techniques including: (1) Dirichlet Multinomial Mixture (DMM) based methods, (2) Global word co-occurrence based methods, and (3) Self-aggregation based methods.

### 3.1 Short Text Topic Modeling Research Context and Evolution

Traditional topic modeling algorithms such as probabilistic latent semantic analysis (PLSA) [8] and latent Dirichlet allocation (LDA) [9] are widely adopted for discovering latent semantic structure from text corpus by capturing word co-occurrence pattern at the document level. Hence, more word co-occurrences would bring in more reliable and better topic inference. Due to the lack of word co-occurrence information in each short text, traditional topic models have a large performance degradation over short texts. Earlier works focus on exploiting external knowledge to help enhance the topic inference of short texts. For example, Phan *et al.* [16] adopted the learned latent topics from Wikipedia to help infer the topic structure of short texts. Similarly, Jin *et al.* [15] searched auxiliary long texts for short texts to infer latent topics of short texts for clustering. A large regular text corpus of high quality is required by these models, which bring in big limitation for these models.

Since 2010, research on topic discovery from short texts has been shifted to merging short texts into long pseudo-documents using different aggregation strategies before adopting traditional topic modeling to infer the latent topics. For example, Weng *et al.* [7] merge all tweets of one user into a pseudo-document before using LDA. Other information

TABLE 3
List of Short Text Topic Modeling Approaches

| Category | Year | Published | Authors | Method | Time Complexity of One Iteration |
|---|---|---|---|---|---|
| DMM | 2014 | KDD [19] | J. Yin & *et al.* | GSDMM | $O(KN\bar{l})$ |
| | 2015 | TACL [29] | D. Nguyen & *et al.* | LF-DMM | $O(O(2KN\bar{l}+KVU))$ |
| | 2016 | SIGIR [30] | C. Li & *et al.* | GPU-DMM | $O(KN\bar{l}+N\bar{l}\zeta+KV)$ |
| | 2017 | TOIS [32] | C. Li & *et al.* | GPU-PDMM | $O(N\bar{l}\sum_{i=1}^{\varsigma-1}C_K^i+N\bar{l}\zeta+KV)$ |
| Global word co-occurrences | 2013 | WWW [14] | X. Chen & *et al.* | BTM | $O(KN\bar{l}c)$ |
| | 2016 | KAIS [33] | Y. Zuo & *et al.* | WNTM | $O(KN\bar{l}c(c-1))$ |
| Self-aggregation | 2015 | IJCAI | X. Quan & *et al.* | SATM | $O(N\bar{l}PK)$ |
| | 2016 | KDD | Y. Zuo & *et al.* | PTM | $O(N\bar{l}(P+K))$ |

includes hashtags, timestamps, and named entities have been tread as metadata to merging short texts [17], [20], [39]. However, helpful metadata may not be accessible in any domains, e.g., news headlines and search snippets. These studies suggest that topic models specifically designed for general short texts are crucial. This survey will provide a taxonomy that captures the existing strategies and these application domains.

## 3.2 A Taxonomy of Short Text Topic Modeling Methods

Below we describe the characteristics of each of these categories and present a summary of some representative methods for each category (cf. Table 3).

## 3.3 Dirichlet Multinomial Mixture Based Methods

Dirichlet Multinomial Mixture model (DMM) was first proposed by Nigam *et al.* [23] based on the assumption that each document is sampled by only one topic. The assumption is more fit for short texts than the assumption that each text is generated by multiple topics. Therefore, many models for short texts were proposed based on this simple assumption [20], [24], [26]. Yin *et al.* [19] proposed a DMM model based on collapse Gibbs sampling. Zhao *et al.* [20] proposed a Twitter-LDA model by assuming that one tweet is generated from one topic. Pretrained word embeddings learned from a very large text corpus are useful because they encode both syntactic and semantic information of words into continuous vectors and similar words are close in vector space. Recently, more work incorporates word embeddings into DMM [30], [32]. Two words with high similarity have a high probability to be put into the same topic, even if they share very limited or no co-occurrences in the current collection of short texts being modeled.

### 3.3.1 GSDMM

DMM respectively chooses Dirichlet distribution for topic-word distribution $\phi$ and document-topic distribution $\theta$ as prior distribution with parameter $\alpha$ and $\beta$. DMM samples a topic $z_d$ for the document $d$ by Multinomial distribution $\theta$, and then generates all words in the document $d$ from topic $z_d$ by Multinomial distribution $\phi_{z_d}$. The graphical model of DMM is shown in Fig. 1. The generative process for DMM is described as follows:

1) Sample a topic proportion $\theta \sim Dirichlet(\alpha)$.
2) For each topic $k \in \{1, \ldots, K\}$:
   Draw a topic-word distribution $\theta_k \sim Dirichlet(\beta)$.
3) For each document $d \in \boldsymbol{D}$:
   (a) Sample a topic $z_d \sim Multinomial(\theta)$.
   (b) For each word $w \in \{w_{d,1}, \ldots, w_{d,n_d}\}$:
       Sample a word $w \sim Multinomial(\phi_{z_d})$.

Gibbs sampling algorithm for Dirichlet Multinomial Mixture model is denoted as GSDMM, which is based on the assumption that each text is sampled by a single topic [19]. Here, for better representation of latent topics, we represent a topic with the topic feature (CF) vector, which essentially is a big document combined with its documents.

The TF vector of a topic $k$ is defined as a tuple $\{n_k^w(w \in W), m_k, n_k\}$, where $n_k^w$ is the number of word $w$ in topic $k$, $m_k$ is the number of documents in topic $k$, and $n_k$ is the number of words in topic $k$.

The topic feature (TF) presents important addible and deletable properties, as described next.

(1) *Addible Property*. A document $d$ can be efficiently added to topic $k$ by updating its TF vector as follows:

$$n_k^w = n_k^w + n_d^w \quad \textit{for each word } w \textit{ in } d$$
$$m_k = m_k + 1 \ ; \ n_k = n_k + n_d.$$

(2) *Deletable Property*. A document $d$ can be efficiently deleted from topic $k$ by updating its TF vector as follows:

$$n_k^w = n_k^w - n_d^w \quad \textit{for each word } w \textit{ in } d$$
$$m_k = m_k - 1 \ ; \ n_k = n_k - n_d.$$

The hidden multinomial variable ($z_d$) for document $d$ is sampled based on collapsed Gibbs sampling, conditioned on a complete assignment of all other hidden variables. GSDMM uses the following conditional probability distribution to infer its topic

$$p(z_d = k | \boldsymbol{Z}_{\neg d}, \boldsymbol{D}) \propto \frac{m_{k,\neg d} + \alpha}{N - 1 + K\alpha} \frac{\prod_{w \in d} \prod_{j=1}^{n_d^w} (n_{k,\neg d}^w + \beta + j - 1)}{\prod_{i=1}^{n_d} (n_{k,\neg d} + V\beta + i - 1)}, \tag{1}$$

where $Z$ represents all topics of all documents, the subscript $\neg d$ means document $d$ is removed from its current topic

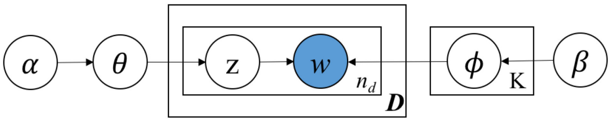


Fig. 1. Graphical model of GSDMM.

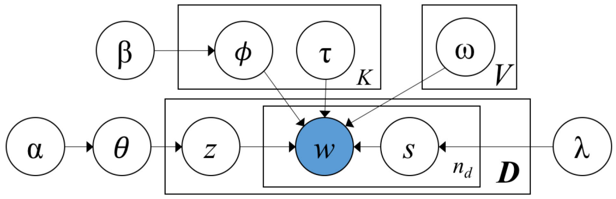


Fig. 2. Graphical model of LF-DMM.

feature (TF) vector, which is useful for the update learning process of GSDMM.

For each document, we first delete it from its current TF vector with the deletable property. Then, we reassign the document to a topic according to the probability of the document belonging to each of the $K$ topics using Equation (1). After obtaining the topic of the document, we add it from its new TF vector with the addible property. Finally, the posterior distribution of each word belonging to each topic is calculated as the follows:

$$\phi_k^w = \frac{n_k^w + \beta}{n_k + V\beta}. \tag{2}$$

### 3.3.2 LF-DMM

The graphical model of LF-DMM is shown in Fig. 2. Based on the assumption that each text is sampled by a single topic, LF-DMM generates the words by Dirichlet multinomial model or latent feature model. Given two latent-feature vectors $\tau$ associated with topic $k$ and $\omega$ associated with word $w$, latent feature model generates a word $w$ using $softmax$ function by the formula

$$\sigma(w \mid \tau_k \boldsymbol{\omega}^T) = \frac{e^{(\tau_k \cdot \omega_w)}}{\Sigma_{w' \in W} e^{(\tau_k \cdot \omega_{w'})}}, \tag{3}$$

where $\boldsymbol{\omega}$ is pre-trained word vectors of all words $W$, and $\omega_w$ is the word vector of word $w$.

For each word $w$ of document $d$, a binary indicator variable $\mathbb{S}_{d,w}$ is sampled from a Bernoulli distribution to determine whether Dirichlet multinomial model or latent feature model will be used to generate $w$. The generative process is described as follows:

1) Sample a topic proportion $\theta \sim Dirichlet(\alpha)$.
2) For each topic $k \in \{1, \ldots, K\}$:
   (a) Draw a topic-word distribution $\theta_k \sim Dirichlet(\beta)$.
3) For each document $d \in \boldsymbol{D}$:
   (a) Sample a topic $z_d \sim Multinomial(\theta)$.
   (b) For each word $w \in \{w_{d,1}, \ldots, w_{d,n_d}\}$:
      (i) Sample a binary indicator variable $\mathbb{S}_{d,w} \in Bernoulli(\lambda)$
      (ii) Sample a word $w \sim (1 - s_w) Multinomial (\phi_{z_d}) + s_w(\sigma(\tau_{z_d}\boldsymbol{\omega}^T))$.

Here, the hyper-parameter $\lambda$ is the probability of a word being generated by latent feature model, and $\mathbb{S}_{d,w}$ indicates whether Dirichlet multinomial model or latent feature model is applied to word $w$ of document $d$. The topic feature (TF) in LF-DMM is similar with GSDMM, so we do not present the addible and deletable properties for LF-DMM.

Based on collapsed Gibbs sampling, LF-DMM uses the following conditional probability distribution to infer the topic of the document $d$

$$\begin{aligned} p(z_d = k | \boldsymbol{Z}_{\neg d}, \boldsymbol{D}, \boldsymbol{\tau}, \boldsymbol{\omega}) \propto (m_{k,\neg d} + \boldsymbol{\alpha}) \\ \prod_{w \in d} ((1-\lambda) \frac{n_{k,\neg d}^w + \beta}{n_{k,\neg d} + V\beta} + \lambda \sigma(w | \tau_k \boldsymbol{\omega}^T))^{n_d^w}, \end{aligned} \tag{4}$$

where $n_d^w$ is the number of word $w$ in document $d$.

The binary indicator variable $\mathbb{S}_{d,w}$ for word $w$ in document $d$ conditional on $z_d = k$ is inferred using the following distribution:

$$p(\mathbb{S}_{d,w} = s | z_d = k) \propto \begin{cases} (1-\lambda) \frac{n_{k,\neg d}^{w_i} + \beta}{n_{k,\neg d} + V\beta} & for\ s = 0, \\ \lambda \sigma(w_i | \tau_k \boldsymbol{\omega}^T) & for\ s = 1. \end{cases} \tag{5}$$

where the subscript $\neg d$ means document $d$ is removed from its current topic feature (TF) vector.

After each iteration, LF-DMM estimates the topic vectors using the following optimization function:

$$\begin{aligned} L_k = -\sum_{w \in W} F_k^w \left( \tau_k \cdot \omega_w - \log \left( \sum_{w' \in W} e^{\tau_k \cdot \omega_{w'}} \right) \right) \\ + \mu ||\tau_k||_2^2, \end{aligned} \tag{6}$$

where $F_k^w$ is the number of times word $w$ generated from topic $k$ by latent feature model. LF-DMM adopted L-BFGS[1] [41] to find the topic vector $\tau_k$ that minimizes $L_k$.

### 3.3.3 GPU-DMM

Based on DMM model, GPU-DMM [30] promotes the semantically related words under the same topic during the sampling process by the generalized Pólya urn (GPU) model [42]. When a ball of a particular color is sampled, a certain number of balls of similar colors are put back along with the original ball and a new ball of that color. In this case, sampling a word $w$ in topic $k$ not only increases the probability of $w$ itself under topic $k$, but also increases the probability of the semantically similar words of word $w$ under topic $k$.

Given pre-trained word embeddings, the semantic similarity between two words $w_i$ and $w_j$ is denoted by $cos(w_i, w_j)$ that are measured by cosine similarity. For all word pairs in vocabulary, if the semantic similarity score is higher that a predefined threshold $\epsilon$, the word pair is saved into a matric $\mathbb{M}$, i.e., $\mathbb{M} = \{(w_i, w_j) | cos(w_i, w_j) > \epsilon\}$. Then, the promotion matrix $\mathbb{A}$ with respect to each word pair is defined below:

$$\mathbb{A}_{w_i, w_j} = \begin{cases} 1 & w_i = w_j \\ \mu & w_j \in \mathbb{M}_{w_i}\ and\ w_j \neq w_i\ , \\ 0 & otherwisex \end{cases} \tag{7}$$

where $\mathbb{M}_{w_i}$ is the row in $\mathbb{M}$ corresponding to word $w_i$ and $\mu$ is the pre-defined promotion weight.

GPU-DMM and DMM share the same generative process and graphical representation but differ in the topic inference process that they use. Different from DMM and LF-DMM, GPU-DMM first samples a topic for a document, and then only reinforces only the semantically similar words if and only if a word has strong ties with the sampled topic. Therefore, a nonparametric probabilistic sampling process

[1] LF-DMM used the implementation of the Mallet toolkit [40]

for word $w$ in document $d$ is as follows:

$$\mathbb{S}_{d,w} \sim Bernoulli(\lambda_{w,z_d}) \tag{8}$$

$$\lambda_{w,z_d} = \frac{p(z|w)}{p_{max}(z'|w)} \tag{9}$$

$$p_{max}(z'|w) = \max_k p(z = k|w)$$

$$p(z = k|w) = \frac{p(z = k)p(w|z = k)}{\sum_{i=1}^{K} p(z = i)p(w|z = i)}, \tag{10}$$

where $\mathbb{S}_{d,w}$ indicates whether GPU is applied to word $w$ of document $d$ given topic $z_d$. We can see that GPU model is more likely to be applied to $w$ if word $w$ is highly relate to topic $z_d$.

The Topic feature vector of a topic $k$ in GPU-DMM is defined as a tuple $\{\widetilde{n}_k^w(w \in W), m_k, \widetilde{n}_k\}$.

TF makes the same changes with GSDMM when no GPU is applied, namely $\mathbb{S}_{d,w} = 0$. Under $\mathbb{S}_{d,w} = 1$, the addible and deletable properties of topic feature (TF) in GPU-DMM are described below.

1) *Addible Property*. A document $d$ will be added into topic $k$ by updating its TF vector as follows:

$$\begin{aligned}
&\widetilde{n}_k = \widetilde{n}_k + n_d^{w_i} \cdot \mathbb{A}_{w_i,w_j} \quad \textit{for each word } w_j \in \mathbb{M}_{w_i} \\
&\widetilde{n}_k^{w_j} = \widetilde{n}_k^{w_j} + n_d^{w} \cdot \mathbb{A}_{w_i,w_j} \quad \textit{for each word } w_j \in \mathbb{M}_{w_i} \\
&m_k = m_k + 1.
\end{aligned}$$

2) *Deletable Property*. A document $d$ will be deleted from topic $k$ by updating its TF vector as follows:

$$\begin{aligned}
&\widetilde{n}_k = \widetilde{n}_k - n_d^{w_i} \cdot \mathbb{A}_{w_i,w_j} \quad \textit{for each word } w_j \in \mathbb{M}_{w_i} \\
&\widetilde{n}_k^{w_j} = \widetilde{n}_k^{w_j} - n_d^{w} \cdot \mathbb{A}_{w_i,w_j} \quad \textit{for each word } w_j \in \mathbb{M}_{w_i} \\
&m_k = m_k - 1.
\end{aligned}$$

Accordingly, based on Gibbs sampling, the conditional distribution to infer the topic for each document in Equation (1) is rewritten as follows:

$$\begin{aligned}
p(z_d = k|\boldsymbol{Z}_{\neg d}, \boldsymbol{D}) \propto \frac{m_{k,\neg d} + \alpha}{N - 1 + K\alpha} \times \\
\frac{\prod_{w\in d}\prod_{j=1}^{n_d^w}(\widetilde{n}_{k,\neg d}^w + \beta + j - 1)}{\prod_{i=1}^{n_d}(\widetilde{n}_{k,\neg d} + V\beta + i - 1)}.
\end{aligned} \tag{11}$$

During each iteration, GPU-PDMM first delete it from its current TF vector with the deletable property. After obtaining the topic of the document, GPU-DMM first updates $\mathbb{S}_{d,w}$ for GPU using Equation (8), and then updates TF vector for each word using the addible property. Finally, the posterior distribution in Equation (2) for GPU-DMM is rewritten as follows:

$$\phi_k^w = \frac{\widetilde{n}_k^w + \beta}{\widetilde{n}_k + V\beta}. \tag{12}$$

### 3.3.4 GPU-PDMM

Considering the single-topic assumption may be too strong for some short text corpus, Li *et al.* [32] first proposed Poisson-based Dirichlet Multinomial Mixture model (PDMM) that allows each document can be generated by one or more (but not too many) topics. Then PDMM can be extended as GPU-PDMM model by incorporating generalized Pólya urn (GPU) model during the sampling process.

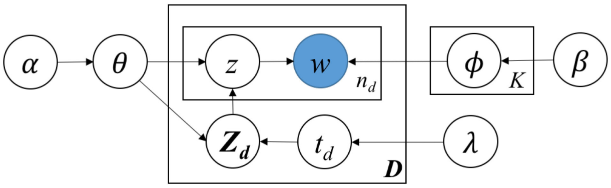


Fig. 3. Graphical model of GPU-PDMM.

In GPU-PDMM, each document is generated by $t_d$ ($0 < t_d \leq \varsigma$) topics, where $\varsigma$ is the maximum number of topics allowable in a document. GPU-PDMM uses Poisson distribution to model $t_d$. The graphical model of GPU-PDMM is shown in Fig. 3. The generative process of GPU-PDMM is described as follows.

1) Sample a topic proportion $\theta \sim Dirichlet(\alpha)$.
2) For each topic $k \in \{1, \ldots, K\}$:
   (a) Draw a topic-word distribution $\theta_k \sim Dirichlet(\beta)$.
3) For each document $d \in \boldsymbol{D}$:
   (a) Sample a topic number $t_d \sim Poisson(\lambda)$.
   (b) Sample $t_d$ distinct topics $\mathbf{Z}_d \sim Multinomial(\theta)$.
   (c) For each word $w \in \{w_{d,1}, \ldots, w_{d,n_d}\}$:
      (i) Uniformly sample a topic $z_{d,w} \sim \mathbf{Z}_d$.
      (ii) Sample a word $w \sim Multinomial(\phi_{z_{d,w}})$.

Here $t_d$ is sampled using Poisson distribution with parameter $\lambda$, and $\mathbf{Z}_d$ is the topic set for document $d$.

The topic feature (TF) vector of a topic $k$ in GPU-PDMM is defined as a tuple $\widetilde{n}_k, \widetilde{n}_k^w(w \in W), c_k, d_k, n_{k,d}, n_{k,d}^w\}$, where $c_k$ is the number of words associated with topic $k$ and $d_k$ represents the word set in topic $k$.

The addible and deletable properties of topic feature (TF) in GPU-PDMM are described below. The $\widetilde{n}_k$ and $\widetilde{n}_k^w$ of TF in GPU-PDMM makes the same changes with GPU-DMM. Here, we only describe other variables in TF.

(1) *Addible Property*. Suppose that word $w$ in document $d$ will be added to topic $k$, TF feature is updates as follows:

$$\begin{aligned}
&c_k = c_k + 1\ ;\ d_k = d_k + w \\
&n_{k,d} = n_{k,d} + 1\ ;\ n_{k,d}^w = n_{k,d}^w + 1.
\end{aligned}$$

(2) *Deletable Property*. Suppose that word $d$ will be deleted from topic $k$. TF feature is updated as follows:

$$\begin{aligned}
&c_k = c_k - 1\ ;\ d_k = d_k - w \\
&n_{k,d} = n_{k,d} - 1\ ;\ n_{k,d}^w = n_{k,d}^w - 1.
\end{aligned}$$

The Gibbs sampling process of GPU-PDMM is similar to GPU-DMM, it updates the topic for word $w$ in document $d$ using the following equation:

$$p(z_{d,w} = k|z_{\neg(d,w)}, \mathbf{Z}_d, \boldsymbol{D}) \propto \frac{1}{t_d} \times \frac{\widetilde{n}_{k,\neg(d,w)}^w + \beta}{\sum_w^V \widetilde{n}_{k,\neg(d,w)}^w + V\beta}. \tag{13}$$

Conditioned on all $z_{d,w}$ in document $d$, GPU-PDMM samples each possible $\mathbf{Z}_d$ as follows:

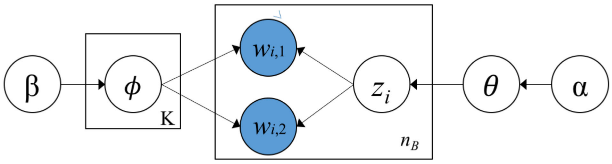


Fig. 4. Graphical model of BTM.

$$p(\mathbf{Z}_d|Z_{\neg d},D) \propto \frac{\lambda^{t_d}}{t_d^{n_d}} \times \frac{\prod_{k\in\mathbf{Z}_d}(c_{k,\neg d}+\alpha)}{\prod_{i=0}^{t_d-1}(\sum_k^K c_{k,\neg d}+K\alpha-i)} \times \prod_{k\in\mathbf{Z}_d}\frac{\prod_{w\in d_k}\prod_{i=0}^{n_{k,d}^w}(\widetilde{n}_{k,\neg d}+n_{k,d}^w)-i+\beta}{\prod_{i=0}^{n_{k,d}-1}(\sum_w^V\widetilde{n}_{k,\neg d}^w+n_{k,d}-i+V\beta)}. \tag{14}$$

During each iteration, for each document $d$, GPU-PDMM first updates TF vector using Deletable Property and the topic for each word $w$ in $d$ using Equation (13). Then GPU-PDMM samples each possible $\mathbf{Z}_d$ using Equation (14). Finally, GPU-PDMM sets all the values of $z_{d,w}$ based on the updates $\mathbf{Z}_d$, updates $\mathbb{S}_{d,w}$ for GPU using Equation (8), and then updates TF vector for each word using the addible property.

Here, due to the computational costs involved in sampling $\mathbf{Z}_d$, GPU-PDMM only samples the more relevant topics for each document. Specifically, GPU-PDMM infers the topic probability $p(z|d)$ of each document $d$ using the follows:

$$p(z=k|d) \propto \sum_{w\in d} p(z=k|w)p(w|d),$$

where $p(w|d)=\frac{n_d^w}{n_d}$. GPU-PDMM only chooses the top $M$ topics for document $d$ based on the probability $p(z|d)$ to generate $\mathbf{Z}_d$, where $\varsigma < M \leq K$. The topic-word distribution can be calculated by Equation (12).

## 3.4 Global Word Co-Occurrences Based Methods

The closer the two words, the more relevance the two words. Utilizing this idea, global word co-occurrences based methods learn the latent topics from the global word co-occurrences obtained from the original corpus. This type of methods needs to set sliding window for extracting word co-occurrences. In general, if the average length of each document is larger than 10, they use sliding window and set the size of the sliding window as 10, else they can directly take each document as a sliding window.

### *3.4.1 BTM*

BTM [14] learns topics over short texts by directly modeling the generation of biterms in the corpus $D$, where a biterm is an unordered word-pair co-occuring in a short context (e.g., a small, fixed-size window over a term sequence within a document). Suppose that the corpus $D$ contains $n_b$ biterms $\mathbf{B}=\{b_i\}_{i=1}^{n_B}$, where $b_i=(w_{i,1},w_{i,2})$. BTM infers topics over the biterms $B$. The generative process of BTM is described as follows, and its graphical model is shown in Fig. 4.

(1) Draw $\theta \sim$ Dirichlet($\alpha$).
(2) For each topic $k \in [1,K]$
  (a) draw $\phi_k \sim$ Dirichlet($\beta$).
(3) For each biterm $b_i \in \mathbf{B}$
  (a) draw $z_i \sim$ Multinomial($\theta$),
  (b) draw $w_{i,1}, w_{i,2} \sim$ Multinomial($\phi_{z_i}$).

The TF vector of a topic $k$ in BTM is defined as a tuple $\{n_k(w \in W), n_k\}$. The addible and deletable properties of topic feature (TF) in BTM are described below.

(1) *Addible Property*. A biterm $b_i$ can be efficiently added to topic $k$ by updating its TF vector as follows:

$$n_k^{w_{i,1}} = n_k^{w_{i,1}}+1\ ;\ n_k^{w_{i,2}} = n_k^{w_{i,2}}+1\ ;\ n_k = n_k+1.$$

(2) *Deletable Property*. A biterm $b_i$ can be efficiently deleted from topic $k$ by unpdating its TF vector as follows:

$$n_k^{w_{i,1}} = n_k^{w_{i,1}}-1\ ;\ n_k^{w_{i,2}} = n_k^{w_{i,2}}-1\ ;\ n_k = n_k-1.$$

Using the technique of collapsed Gibbs sampling, BTM samples the topic $z_i$ of biterm $b_i$ using the following conditional distribution:

$$p(z_i=k|\mathbf{Z}_{\neg i},\mathbf{B}) \propto (n_{k,-i}+\alpha)\times \frac{(n_{k,\neg i}^{w_{i,1}}+\beta)(n_{k,\neg i}^{w_{i,2}}+\beta)}{(n_{k,\neg i}+V\beta+1)(n_{k,\neg i}+V\beta)}, \tag{15}$$

where $Z_{\neg i}$ denotes the topics for all biterms except the current biterm $b_i$, and $n_k$ is the number of biterms assigned to topic $k$.

For each biterm, we first delete it from its current TF vector with the deletable property. Then, we reassign the biterm to a topic using Equation (4). Accordingly, we update the new TF vector with the addible property. After finishing the iterations, BTM estimates $\phi$ and $\theta$ as follows:

$$\phi_k^w = \frac{n_k^w+\beta}{n_k+V\beta}, \tag{16}$$

$$\theta_d^k = \sum_{i=1}^{n_d^b} p(z_i=k), \tag{17}$$

where $\theta_d^k$ is the probability of topic $k$ in document $d$, and $n_d^b$ is the number of biterms in document $d$.

### *3.4.2 WNTM*

WNTM [33] uses global word co-occurrence to construct word co-occurrence network, and learns the distribution over topics for each word from word co-occurrence network using LDA. WNTM first set the size of a sliding window, and the window is moving word by word. Suppose window size is set as 10 in the original paper, if one document has 15 words, it will have 16 windows in this document. As WNTM scanning word by word in one window, two distinct words in the window are regarded as co-occurrence. WNTM construct undirected word co-occurrence network, where each node of the word co-occurrence network represents one word and the weight of each edge is the number of co-occurrence of the two connected words. We can see that the number of nodes is the number of vocabulary $V$.

Then, WNTM generates one pseudo-document $l$ for each vertex $v$ which is consisted of the adjacent vertices of this vertex in word network. The occur times of this adjacent vertex in $l$ is determined by the weight of the edge. The number of words in $l$ is the degree of the vertex $v$ and the number of pseudo-documents $P$ is the number of vertices.

After obtaining pseudo-documents $P$, WNTM adopts LDA to learn latent topics from pseudo-documents. Therefore, the topic feature (TF) in LF-DMM is same with LDA. For each word $w$ in $l$, WNTM infers its topic using the following conditional distribution:

$$p(z_{l,w} = k \mid Z_{\neg(l,w)}, P, \alpha, \beta) \propto (n^k_{l,\neg(l,w)} + \alpha) \frac{n^w_{k,\neg(l,w)} + \beta}{n_{k,\neg(l,w)} + V\beta}, \tag{18}$$

where $n^k_l$ is the number of topic $k$ belonging to pseudo-document $l$, and $\neg(l,w)$ means word $w$ is removed from its pseudo-document $l$.

Because each pseudo-document is each word's adjacent word-list, the document-topic distribution learned from pseudo-document is the topic-word distribution in WNTM. Suppose pseudo-document $l$ is generated from word $w$, the topic-word distribution of $w$ is calculated using the following Equation:

$$\phi^w_k = \frac{n^k_l + \alpha}{n_l + K\alpha}, \tag{19}$$

where $n_l$ is the number of words in $l$.

Given topic-word distribution, the document-word distribution $\theta_d$ can be calculated as

$$\theta^k_d = \sum_{i=1}^{n_d} \phi_k^{w_{d,i}} p(w_{d,i}|d)$$

$$p(w_{d,i}|d) = \frac{n_d^{w_{d,i}}}{n_d},$$

where $n_d^{w_{d,i}}$ is the number of word $w_{d,i}$ in document $d$.

## 3.5 Self-Aggregation Based Methods

Self-aggregation based methods alleviate the problem of sparseness by merging short texts into long pseudo-documents $P$ before inferring the latent topics [15], [17], [37]. The previous self-aggregation based methods first merged short texts, and then applied topic models. Recently, SATM and PTM simultaneously integrate clustering and topic modeling in one iteration. In general, the number of pseudo-documents $|P|$ is significantly less than the number of short texts, namely $|P| \ll N$.

### 3.5.1 SATM

Self-aggregation based topic modeling (SATM) [21] supposes that each short text is sampled from an unobserved long pseudo-document, and infers latent topics from pseudo-documents using standard topic modeling. The Gibbs sampling process in SATM can be described in two indispensable steps.

The first step calculates the probability of the occurrence of a pseudo-document $l$ in $P$ conditioned on short document $d$ in short corpus, which is estimated using the mixture of unigrams model [23],

w˙d,i

$$p(l|d) = \frac{p(l) \prod_{i=1}^{V} (\frac{n_l^{w_{d,i}}}{n_l})^{n_d^{w_{d,i}}}}{\sum_{m=1}^{|P|} p(m) \prod_{i=1}^{V} (\frac{n_m^{w_{d,i}}}{n_m})^{n_d^{w_{d,i}}}}, \tag{20}$$

where $p(l) = \frac{n_l}{N}$ represents the probability of pseudo-document $p_l$, $n_l^{w_{d,i}}$ is the number of word $w_{d,i}$ in pseudo-document $p_l$, and $n_l$ is the number of words in $p_l$.

The second step estimates draws a pair of pseudo-document label $l_{d,w}$ and topic label $z_{d,w}$ jointly for word $w$ in document $d$, which is similar with standard topic modeling (author-topic modeling) [43].

The addible and deletable properties of pseudo-document and topic feature (PTF) in SATM are described below.

(1) *Addible Property*. A word $w$ can be efficiently added into pseudo-document $l$ and topic $k$ by updating its TPF vector as follows:

$$\begin{aligned} n^w_l &= n^w_l + 1 \ ; \ n^k_l = n^k_l + 1 \ ; \ n_l = n_l + 1 \\ n^w_k &= n^w_k + 1 \ ; \ n_k = n_k + 1. \end{aligned}$$

(2) *Deletable Property*. A word $w$ can be efficiently deleted from pseudo-document $l$ and topic $k$ by updating its PTF vector as follows:

$$\begin{aligned} n^w_l &= n^w_l - 1 \ ; \ n^k_l = n^k_l - 1 \ ; \ n_l = n_l - 1 \\ n^w_k &= n^w_k - 1 \ ; \ n_k = n_k - 1. \end{aligned}$$

The pair of pseudo-document label $l_{d,w}$ and topic label $z_{d,w}$ jointly for word $w$ in document $d$ can be calculated by

$$\begin{aligned} & p(l_{d,w} = l, z_{d,w} = k | \mathbf{Z}_{\neg(d,w)}, P_{\neg(d,w)}) \propto \\ & \qquad p(l|d) \times \frac{n^k_{l,\neg(d,w)} + \alpha}{n_{l,\neg(d,w)} + K\alpha} \cdot \frac{n^w_{k,\neg(d,w)} + \beta}{n_{k,\neg(d,w)} + V\beta}, \end{aligned} \tag{21}$$

where $n^k_l$ is the number of words in pseudo-document $l$ belonging to topic $k$.

After finishing the iterations, SATM estimates $\phi$ and $\theta$ as follows:

$$\phi^w_k = \frac{n^w_k + \beta}{n_k + V\beta}, \tag{22}$$

$$\theta^k_d = \prod_{i=1}^{n_d} \phi_k^{w_{d,i}}. \tag{23}$$

### 3.5.2 PTM

The pseudo-document-based topic modeling (PTM) [36] supposes that each short text is sampled from one long pseudo-document $p_l$, and then infers the latent topics from long pseudo-documents $P$. A multinomial distribution $\varphi$ is used to model the distribution of short texts over pseudo-documents. The graphical model of PTM is shown in Fig. 5. The generative process of PTM is described as follows,

(1) Sample $\varphi \sim Dir(\lambda)$
(2) For each topic $k \in [1, K]$

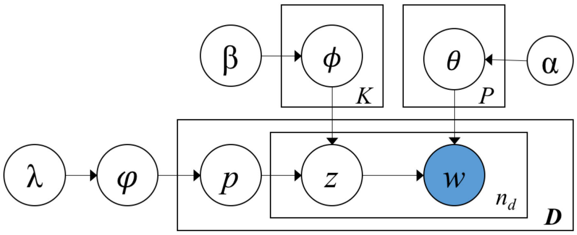


Fig. 5. Graphical model of PTM.

(a) draw $\phi_k \sim$ Dirichlet($\beta$).

(3) For each pseudo-document $l$

(a) sample $\theta_l \sim Dir(\alpha)$

(4) For each document $d \in \boldsymbol{D}$:

(a) Sample a pseudo-document $l \sim Multinomial(\varphi)$.

(b) For each word $w \in \{w_{d,1}, \ldots, w_{d,n_d}\}$ in $d$:

(i) Sample a topic $z \sim Multinomial(\theta_l)$.

(ii) Sample a word $w \sim Multinomial(\phi_z)$.

The addible and deletable properties of pseudo-document and topic feature (PTF) in PTM are described below.

(1) *Addible Property*. A document $d$ can be efficiently added into pseudo-document $l$ by updating its PTF vector as follows:

$$n_l^k = n_l^k + 1 \quad for\ z_{d,w} = k\ in\ d$$
$$m_l = m_l + 1\ ;\ n_l = n_l + n_d.$$

(2) *Deletable Property*. A document $d$ can be efficiently deleted from pseudo-document $l$ by updating its PF vector as follows:

$$n_l^k = n_l^k - 1 \quad for\ z_{d,w} = k\ in\ d$$
$$m_l = m_l - 1\ ;\ n_l = n_l - n_d.$$

Integrating out $\theta, \phi$ and $\varphi$, the pseudo-document assignment $l$ for short text $d$ based on collapsed Gibbs sampling can be estimated as follows:

$$p(l_d = l|\overrightarrow{P_{\neg d}}, \boldsymbol{D}) \propto \frac{m_{l,\neg d}}{N-1+\lambda|P|}\frac{\prod_{k\in d}\prod_{j=1}^{n_d^k}(n_{l,\neg d}^k+\alpha+j-1)}{\prod_{i=1}^{n_d}(n_{l,\neg d}+K\alpha+i-1)}, \qquad (24)$$

where $m_l$ is the number of short texts associated with pseudo-document $l$, $n_l^k$ is the number of words associated with topic $k$ in pseudo-document $l$.

After obtaining the pseudo-document for each short text, PTM samples the topic assignment for each word $w$ in document $d$. That is

$$p(z_{d,w} = k|\mathbf{Z}_{\neg(d,w)}, \boldsymbol{D}) \propto (n_l^k + \alpha)\frac{n_k^w + \beta}{n_k + V\beta}, \qquad (25)$$

where $n_l^k$ is the number of words associated with topic $k$ in pseudo-document $l$.

The document-word distribution $\theta_d$ can be calculated as

$$\theta_d^k = \frac{n_d^k + \alpha}{n_d + K\alpha}. \qquad (26)$$

## 4 Applications

With the emerging of social media, topic models have been used for social media content analysis, such as content characterizing and recommendation [20], [44], text classification [45], [46], event tracking [47], [48], [49], community discovery [50]. However, although the corpus is composed of short texts, some previous work directly applied traditional topic models for topic discovery, since no specific short text topic models were proposed at that time. Therefore, it brings a new chance for short text topic modeling to improve the performance of these tasks.

### 4.1 Content Characterizing and Recommendation

Microblogging sites are used as publishing platforms to create and consume content from sets of users with overlapping and disparate interests, which results in many contents are useless for users. These work [20], [44] have been devoted to content analysis of Twitter. Ramage *et al.* [44] used topic models to discover latent topics from the tweets that can be roughly categorized into four types: substance topics about events and ideas, social topics recognizing language used toward a social end, status topics denoting personal updates, and style topics that embody broader trends in language usage. Next, they characterize selected Twitter users along these learned dimensions for providing interpretable summaries or characterizations of users tweet streams. Zhao *et al.* [20] performed content analysis on tweets using topic modeling to discover the difference between Twitter and traditional medium.

Content analysis is crucial for content recommendation for microblogging users [51], [52]. Phelan *et al.* [53] identified emerging topics of interest from Twitter information using topic modeling, and recommended news by matching emerging topics and recent news coverage in an RSS feed. Chen *et al.* [54] also studied content recommendation based on Twitter for better capture users' attention by exploring three separate dimensions in designing such a recommender: content sources, topic interest models for users, and social voting. Yin *et al.* [55] focused on the problem of dynamic user behavior modeling in social media systems and its applications in temporal recommendations. They proposed a temporal context-aware mixture model (TCAM) that explicitly introduces two types of latent topics (intrinsic interest and the temporal context) to model user interests and temporal context, respectively.

### 4.2 Text Classification

Topic models for text classification are mainly from the following two aspects. The first one is topics discovered from external large-scale data corpora are added into short text documents as external features. For example, Phan *et al.* [16] built a classifier on both a set of labeled training data and a set of latent topics discovered from a large-scale data collection. Chen *et al.* [56] integrated multi-granularity hidden topics discovered from short texts and produced discriminative features for short text classification. Vo *et al.* [57] explored more external large-scale data collections which contain not only Wikipedia but also LNCS and DBLP for discovering latent topics.

The other one is that topic models are used to obtain a low dimensional representation of each text, and then classify text using classification methods [45], [46]. Compared with traditional statistical methods, the representation using topic

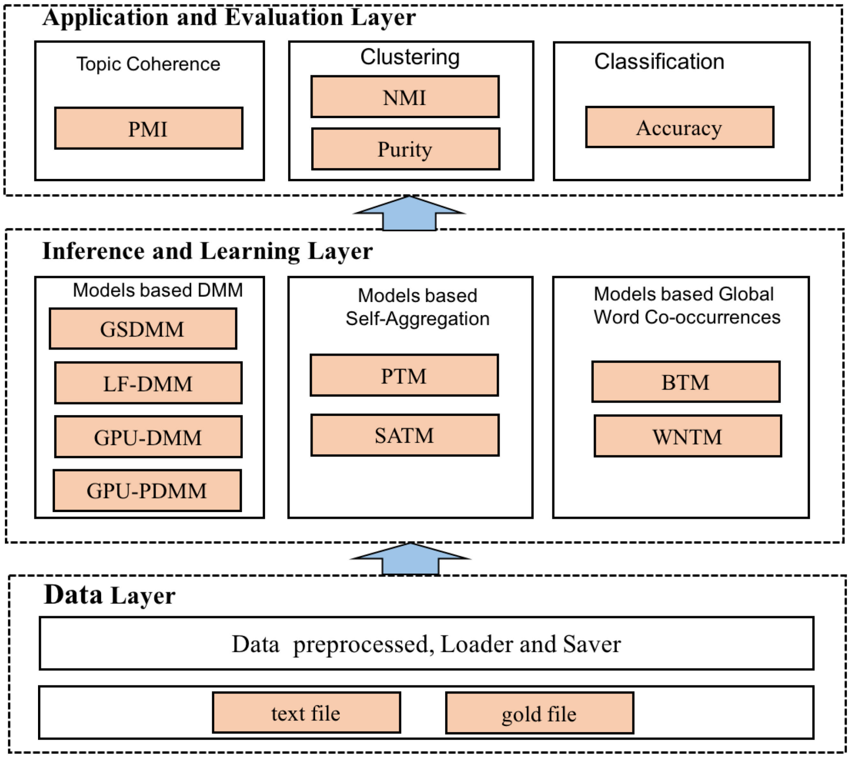

Application and Evaluation Layer
Topic Coherence
PMI
Clustering
NMI
Purity
Classification
Accuracy
Inference and Learning Layer
Models based DMM
GSDMM
LF-DMM
GPU-DMM
GPU-PDMM
Models based Self-Aggregation
PTM
SATM
Models based Global Word Co-occurrences
BTM
WNTM
Data Layer
Data preprocessed, Loader and Saver
text file
gold file

Fig. 6. The architecture of STTM.

models can get a compact, dense and lower dimensional vector in which each dimension of the vector usually represents a specific semantic meaning (e.g., a topic) [23]. Dai *et al.* [45] used the topic information from training data to extend representation for short text. Recent topic modeling methods on text representation have explicitly evaluated their models on this task. They showed that a low dimensional representation for each text suffices to capture the semantic information.

### 4.3 Event Tracking

Nowadays, a large volume of text data is generated from the social communities, such as blogs, tweets, and comments. The important task of event tracking is to observe and track the popular events or topics that evolve over time [47], [48], [49]. Lin *et al.* [47] proposed a novel topic modeling that models the popularity of events over time, taking into consideration the burstiness of user interest, information diffusion in the network structure, and the evolution of latent topics. Lau *et al.* [58] designed a novel topic modeling for event detecting, whose model has an in-built update mechanism based on time slices by implementing a dynamic vocabulary.

For a better tracking topic or event, spatial information is incorporated to infer the latent topics. Yin *et al.* [59] proposed a novel solution to detect both stable and temporal topics simultaneously from social media data by exploiting prior spatial information in a social network.

## 5 A Java Library for Short Text Topic Modeling

We released an open-source Java library, Short Text Topic Modeling (STTM),[2] which is the first comprehensive open-source library, which not only includes the state-of-the-art algorithms with a uniform easy-to-use programming interface but also includes a great number of designed modules for the evaluation and application of short text topic modeling algorithms. The design of STTM follows three basic principles. (1) Preferring integration of existing algorithms rather than implementing them. If the original implementations are open, we always attempt to integrate the original codes rather than implement them. The work that we have done is to consolidate the input/output file formats and package these different approaches into some newly designed java classes with a uniform easy-to-use member functions. (2) Including traditional topic modeling algorithms for long texts. The classical topic modeling algorithm (LDA [9] and its variation LF-LDA [29]) are integrated, which is easy for users to the comparison of long text topic modeling algorithms and short text topic modeling algorithms. (3) Extendibility. Because short text topic modeling is an emerging research field, many topics have not been studied yet. For incorporating future work easily, we try to make the class structures as extendable as possible when designing the core modules of STTM.

Fig. 6 shows the hierarchical architecture of STTM. STTM supports the entire knowledge discovery procedure including analysis, inference, evaluation, application for classification and clustering. In the data layer, STTM is able to read a text file, in which each line represents one document. Here, a document is a sequence of words/tokens separated by whitespace characters. If we need to evaluate the algorithm, we also need to read a gold file, in which each line is the class label of one document. STTM provides implementations of DMM [19], LF-DMM [29], GPU-DMM [30], GPU-PDMM [32], BTM [14], WNTM [33], SATM [21], and PTM [36]. For each model, we not only provide how to train a model on existing corpus but also give how to infer topics

[2] https://github.com/qiang2100/STTM

on a new/unseen corpus using a pre-trained topic model. In addition, STTM presents three aspects of how to evaluate the performance of the algorithms (i.e., topic coherence, clustering, and classification). For topic coherence, we use the point-wise mutual information (PMI) to measure the coherence of topics [36]. Short text topic modeling algorithms are widely used for clustering and classification. In the clustering module, STTM provides two measures (NMI and Purity) [22]. Based on the latent semantic representations learned by short text topic modeling, accuracy metric is chosen in classifications [14].

## 6 Experimental Setup

In this section, we specify the parameter setting of the introduced short text topic models, dataset and evaluation metrics we used. All of these are implemented in our library STTM. The experiments were performed on a Ubuntu 18.04(bionic) system with 6 cores, Intel Xeon E5645 CPU and 12288 KB cache.

For all models in comparison, we used the recommended setting by the authors and set the number of iterations as 2,000 unless explicitly specified elsewhere. The word embeddings of LF-DMM, GPU-DMM, and GPU-PDMM are trained by Glove [60]. In this paper, we used the pre-trained word embeddings "glove.6B.200d.txt", where the dimension of the vector is 200.

### 6.1 Parameter Setting

*LDA*. LDA is the most popular and classic topic modeling. We choose it as a baseline to the comparison. The hyper-parameters of LDA are set as $\alpha = 0.05$ and $\beta = 0.01$ that are proved in the paper (BTM). The authors tuned parameters via grid search on the smallest collection to get the best performance.

*GSDMM*. We set $k = 300$, $\alpha = 0.1$ and $\beta = 0.1$ declared in the paper(GSDMM).

*LF-DMM*. We set $\lambda = 0.6$, $\alpha = 0.1$ and $\beta = 0.01$ shown in their paper. We set the iterations for baseline models as 1,500 and ran the further iterations 500 times.

*GPU-DMM*. We use the hyper-parameter settings provided by the authors, $\alpha = 50/k$, $\beta = 0.01$. The number of iterations is 1,000 in the paper.

*GPU-PDMM*. All the settings are same as the model *GPU-DMM*. We set $\lambda = 1.5$, $\varsigma = 1.5$, and $M$=10 declared in the original paper.

*BTM*. The parameters $\alpha = 50/K$ and $\beta = 0.01$ are used, the model gets optimal performance. Each document is treated as one window.

*WNTM*. We set $\alpha = 0.1$ and $\beta = 0.1$ used in the original paper. The window size is set as 10 words.

*SATM*. We set the number of pseudo-numbers as 300, and the hyper-parameters $\alpha = 50/k$, $\beta = 0.1$. The number of iterations is set as 1,000.

*PTM*. The hyper-parameters are $\alpha = 0.1$ and $\beta = 0.01$. We also set the number of pseudo-document as 1,000.

### 6.2 Datasets

To show the effects and differences of the above nine models, we select the following six datasets to verify the models. After preprocessing these datasets, we present the key information of the datasets that are summarized in Table 4, where $K$ corresponds to the number of topics per dataset, $N$ represents the number of documents in each dataset, $Len$ shows the average length and maximum length of each document, and $V$ indicates the size of the vocabulary.

TABLE 4
The Basic Information of the Corpus

| *Dataset* | *K* | *N* | *Len* | *V* |
|---|---|---|---|---|
| SearchSnippets | 8 | 12,295 | 14.4/37 | 5,547 |
| StackOverflow | 20 | 16,407 | 5.03/17 | 2,638 |
| Biomedicine | 20 | 19,448 | 7.44/28 | 4498 |
| Tweet | 89 | 2,472 | 8.55/20 | 5,096 |
| GoogleNews | 152 | 11,109 | 6.23/14 | 8,110 |
| PascalFlickr | 20 | 4,834 | 5.37/19 | 3,431 |

*SearchSnippets.* Given the predefined phrases of 8 different domains, this dataset was chosen from the results of web search transaction. The 8 domains are Business, Computers, Culture-Arts, Education-Science, Engineering, Health, Politics-Society, and Sports, respectively.

*StackOverflow.* The dataset is released on Kaggle.com. The raw dataset contains 3,370,528 samples from July 31st, 2012 to August 14, 2012. Here, the dataset randomly selects 20,000 question titles from 20 different tags.

*Biomedicine.* Biomedicine makes use of the challenge data delivered on BioASQ's official website.

*Tweet.* In the 2011 and 2012 microblog tracks at Text REtrieval Conference (TREC), there are 109 queries for using. After removing the queries with none highly-relevant tweets, Tweet dataset includes 89 clusters and totally 2,472 tweets.

*GoogleNews.* In the Google news site, the news articles are divided into clusters automatically. GoolgeNews dataset is downloaded from Google news site on November 27, 2013, and crawled the titles and snippets of 11,109 news articles belonging to 152 clusters.

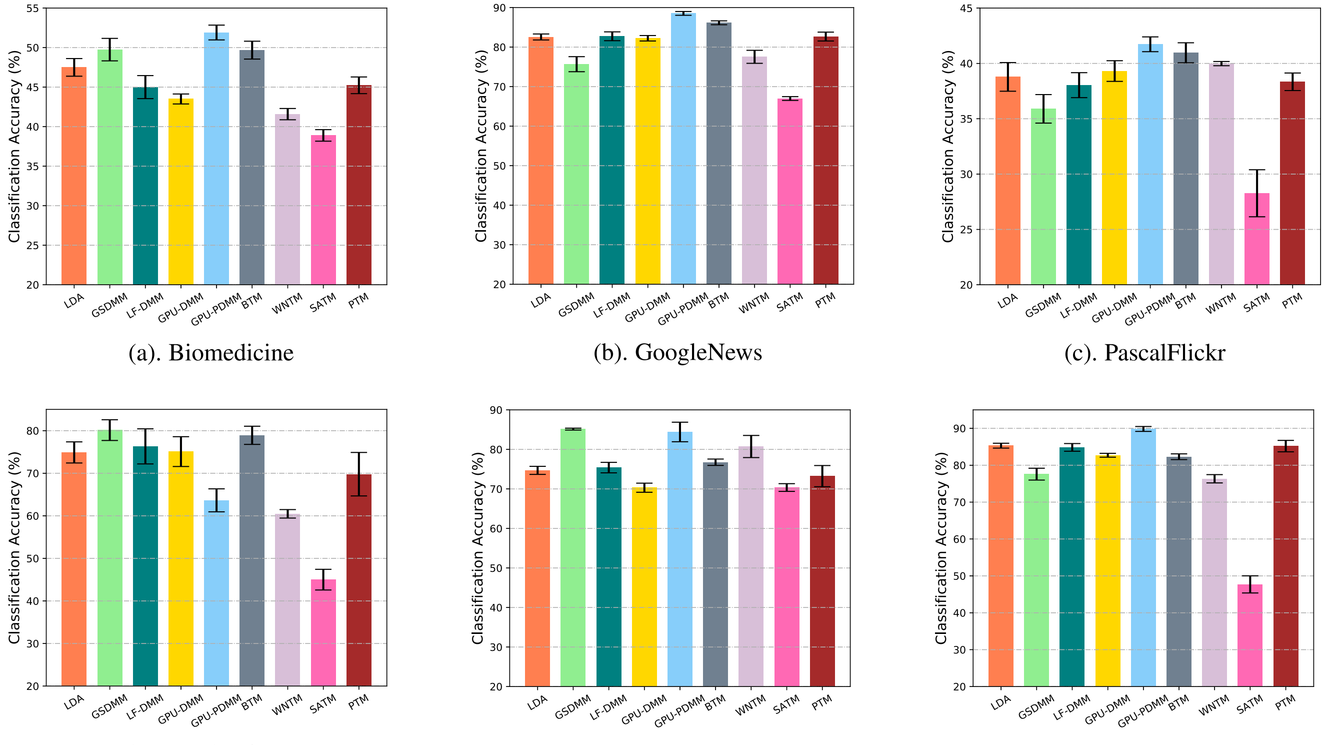
*PascalFlickr.* PascalFlickr dataset are a set of captions [61], which is used as evaluation for short text clustering [62].

### 6.3 Evaluation Metrics

It is still an open problem about how to evaluate short text topic models. A lot of metrics have been proposed for measuring the coherence of topics in texts [63], [64]. Although some metrics tend to be reasonable for long texts, they can be problematic for short texts [21]. Most conventional metrics (e.g., perplexity) try to estimate the likelihood of held-out testing data based on parameters inferred from training data. However, this likelihood is not necessarily a good indicator of the quality of the extracted topics [65]. To provide a good evaluation, we evaluate all models from many aspects using different metrics,

*Classification Evaluation*. Each document can bed represented using document-topic distribution $p(z|d)$. Therefore, we can evaluate the performance of topic modeling using text classification. Here, we choose accuracy as a metric for classification. Higher accuracy means the learned topics are more discriminative and representative. We use a linear kernel Support Vector Machine (SVM) classifier in LIBLINEAR[3] with the

[3] https://liblinear.bwaldvogel.de/



(a). Biomedicine

(b). GoogleNews

(c). PascalFlickr

(d). SearchSnippets

(e). StackOverflow

(f). Tweet

Fig. 7. Average classification accuracy of all models on six datasets.

default parameter settings. The accuracy of classification is computed through fivefold cross-validation on all datasets.

*Cluster Evaluation (Purity and NMI)*. By choosing the maximum of topic probability for each document, we can get the cluster label for each text. Then, we can compare the cluster label and the golden label using metric Purity and NMI [19], [25].

*Topic Coherence*. Computing topic coherence, additional dataset (Wikipedia) as a single meta-document is needed to score word pairs using term co-occurrence in the paper (Automatic Evaluation of Topic Coherence). Here, we calculate the point-wise mutual information (PMI) of each word pair, estimated from the entire corpus of over one million English Wikipedia articles [32]. Using a sliding window of 10 words to identify co-occurrence, we computed the PMI of all a given word pair. The Wikipedia corpus can be downloaded here.[4] Then, we can transfer the dataset from HTML to text using the code in the STTM package. Finally, due to the large size, we only choose 1,000,000 sentences from it.

## 7 Experiments and Analysis

In this section, we conduct experiments to evaluate the performance of the nine models. We run each model 20 times on each dataset and report the mean and standard deviation.

### 7.1 Classification Accuracy

Classification accuracy is used to evaluate document-topic distribution. We represent each document with its document-topic distribution and employ text classification method to assess. For DMM based methods, we use $p(z_d = k)$ to represent each document. For other models, we adopt the giving equation $\theta_d^k$.

The classification accuracy on six datasets using nine models is shown in Fig. 7. We observe that although the performance of methods is dataset dependent, DMM based methods which utilize word embeddings outperform others, especially on Tweet and GoogleNews datasets. This is because GoogleNews and Tweet are general (not domain-specific) datasets and word embeddings used in this paper are trained in general datasets. If we try to use these models (LF-DMM, GPU-DMM, and GPU-PDMM) on domain-specific datasets, we can further improve the performance by pre-trained word embeddings on domain-specific datasets. These observations validate that incorporating general word semantic relations is beneficial for short text topic modeling. But, different strategies about incorporating word embeddings will lead to different results. We observe that both LF-DMM and GPU-DMM have poor results on both datasets (Biomedicine and StackOverflow) even though they also exploit word embeddings. A simple switch mechanism with an indication variable used in LF-DMM may not optimally balance the two components. As to LFDMM, one possible reason for its modest performance is that harnessing the semantic relatedness via a two-component mixture may need a more complex mechanism. As to GPU-DMM, it can obtain 80 percent accuracy with $\alpha = 0.1$ and $\beta = 0.1$ on StackOverflow, which means that GPU-DMM is sensitive to the hyperparameters. This illustrates that the GPU mechanism is an appropriate strategy to exploit word semantic relations offered by word embeddings.

We also observe that self-aggregation based methods are unable to achieve high accuracy, especially the SATM method. The performance of self-aggregation based methods is affected

[4] https://dumps.wikimedia.org/enwiki/

TABLE 5
*Purity* and *NMI* Value of All Models on Six Datasets

| Model | | Biome dicine | Google News | Pascal Flickr | Search Snippets | Stack Overflow | Tweet | Mean Value |
|---|---|---|---|---|---|---|---|---|
| LDA | *Purity* | 0.456 ± 0.011 | 0.793 ± 0.005 | 0.376 ± 0.013 | 0.740 ± 0.029 | 0.562 ± 0.013 | 0.821 ± 0.006 | 0.625 ± 0.013 |
| | *NMI* | 0.356 ± 0.004 | 0.825 ± 0.002 | 0.321 ± 0.006 | 0.517 ± 0.025 | 0.425 ± 0.006 | 0.805 ± 0.004 | 0.542 ± 0.008 |
| GSDMM | *Purity* | **0.494 ± 0.011** | 0.754 ± 0.014 | 0.360 ± 0.012 | **0.801 ± 0.024** | 0.713 ± 0.002 | 0.785 ± 0.011 | 0.650 ± 0.013 |
| | *NMI* | **0.396 ± 0.006** | 0.851 ± 0.004 | 0.317 ± 0.005 | **0.608 ± 0.023** | 0.593 ± 0.002 | 0.801 ± 0.007 | **0.590 ± 0.001** |
| LF-DMM | *Purity* | 0.421 ± 0.019 | 0.828 ± 0.009 | 0.381 ± 0.009 | 0.762 ± 0.042 | 0.518 ± 0.0217 | 0.856 ± 0.009 | 0.630 ± 0.018 |
| | *NMI* | 0.348 ± 0.005 | 0.875 ± 0.005 | 0.365 ± 0.007 | 0.579 ± 0.026 | 0.443 ± 0.007 | 0.843 ± 0.006 | 0.578 ± 0.009 |
| GPU-DMM | *Purity* | 0.433 ± 0.008 | 0.818 ± 0.005 | **0.395 ± 0.010** | 0.751 ± 0.035 | 0.511 ± 0.013 | 0.830 ± 0.006 | 0.623 ± 0.013 |
| | *NMI* | 0.366 ± 0.006 | 0.852 ± 0.002 | **0.370 ± 0.004** | 0.561 ± 0.026 | 0.429 ± 0.003 | 0.810 ± 0.006 | 0.565 ± 0.008 |
| GPU-PDMM | *Purity* | 0.481 ± 0.011 | **0.860 ± 0.002** | 0.368 ± 0.008 | 0.537 ± 0.030 | 0.702 ± 0.032 | **0.869 ± 0.005** | 0.636 ± 0.015 |
| | *NMI* | 0.381 ± 0.005 | 0.871 ± 0.001 | 0.322 ± 0.003 | 0.341 ± 0.014 | 0.607 ± 0.013 | 0.830 ± 0.003 | 0.559 ± 0.007 |
| BTM | *Purity* | 0.458 ± 0.012 | 0.849 ± 0.005 | 0.392 ± 0.011 | 0.765 ± 0.032 | 0.537 ± 0.019 | 0.814 ± 0.008 | 0.636 ± 0.014 |
| | *NMI* | 0.380 ± 0.004 | 0.875 ± 0.003 | 0.368 ± 0.006 | 0.566 ± 0.027 | 0.456 ± 0.008 | 0.808 ± 0.005 | 0.575 ± 0.009 |
| WNTM | *Purity* | 0.472 ± 0.009 | 0.837 ± 0.007 | 0.324 ± 0.005 | 0.712 ± 0.016 | **0.750 ± 0.026** | 0.856 ± 0.012 | **0.658 ± 0.013** |
| | *NMI* | 0.369 ± 0.004 | **0.876 ± 0.004** | 0.295 ± 0.003 | 0.464 ± 0.011 | **0.659 ± 0.006** | **0.850 ± 0.009** | 0.585 ± 0.006 |
| SATM | *Purity* | 0.384 ± 0.007 | 0.654 ± 0.008 | 0.237 ± 0.059 | 0.459 ± 0.055 | 0.505 ± 0.019 | 0.392 ± 0.011 | 0.438 ± 0.027 |
| | *NMI* | 0.27 ± 0.001 | 0.76 ± 0.005 | 0.186 ± 0.049 | 0.205 ± 0.036 | 0.366 ± 0.011 | 0.507 ± 0.006 | 0.382 ± 0.018 |
| PTM | *Purity* | 0.425 ± 0.012 | 0.807 ± 0.010 | 0.359 ± 0.012 | 0.674 ± 0.057 | 0.481 ± 0.034 | 0.839 ± 0.007 | 0.597 ± 0.022 |
| | *NMI* | 0.353 ± 0.003 | 0.866 ± 0.005 | 0.336 ± 0.010 | 0.457 ± 0.045 | 0.442 ± 0.016 | 0.846 ± 0.006 | 0.550 ± 0.014 |

by generating long pseudo-documents. Without any auxiliary information or metadata, the error of this step of generating pseudo-documents will be amplified in the next step.

In conclusion, these models based on the simple assumption (BTM and GSDMM) always outperform than LDA in all datasets, which indicate that two words or all words in one document are very likely to from one topic. Here we can see that the performance of other models (LF-DMM, GPU-DMM, GPU-PDMM, WNTM) are highly data set dependent. For example, WNTM achieves good performance on Tweet, GoogleNews and StackOverflow, but performs poorly on other data sets. GPU-PDMM achieves the best performance on all data sets, except SearchSnippets.

### 7.2 Clustering

Another important application of short text topic modeling is short text clustering. For each document, we choose the maximum value from its topic distribution as the cluster label. We report the mean value of each modeling on all datasets in the last column. The best results for each dataset using each metric are highlighted in bold.

Table 5 illustrates the results using cluster metrics. We can see that all models outperform long text topic modeling (LDA), except SATM. Here, similar to the conclusions in classification, we can observe that the performance of approaches is highly data set dependent. WNTM achieves the best performance on several datasets but performs poorly on PascalFlickr. GPU-PDMM performs very well on all datasets except SearchSnippets.

For self-aggregation based methods, PTM performs better than SATM. For global word-occurrences based methods, two methods are very different from each other. WNTM performs better than BTM on Tweet and StackOverflow, and BTM achieves good performance on GoogleNews and PascalFlickr. For DMM based methods, GSDMM without incorporating word embeddings outperforms other methods on Biomedicine and SearchSnippets.

### 7.3 Topic Coherence

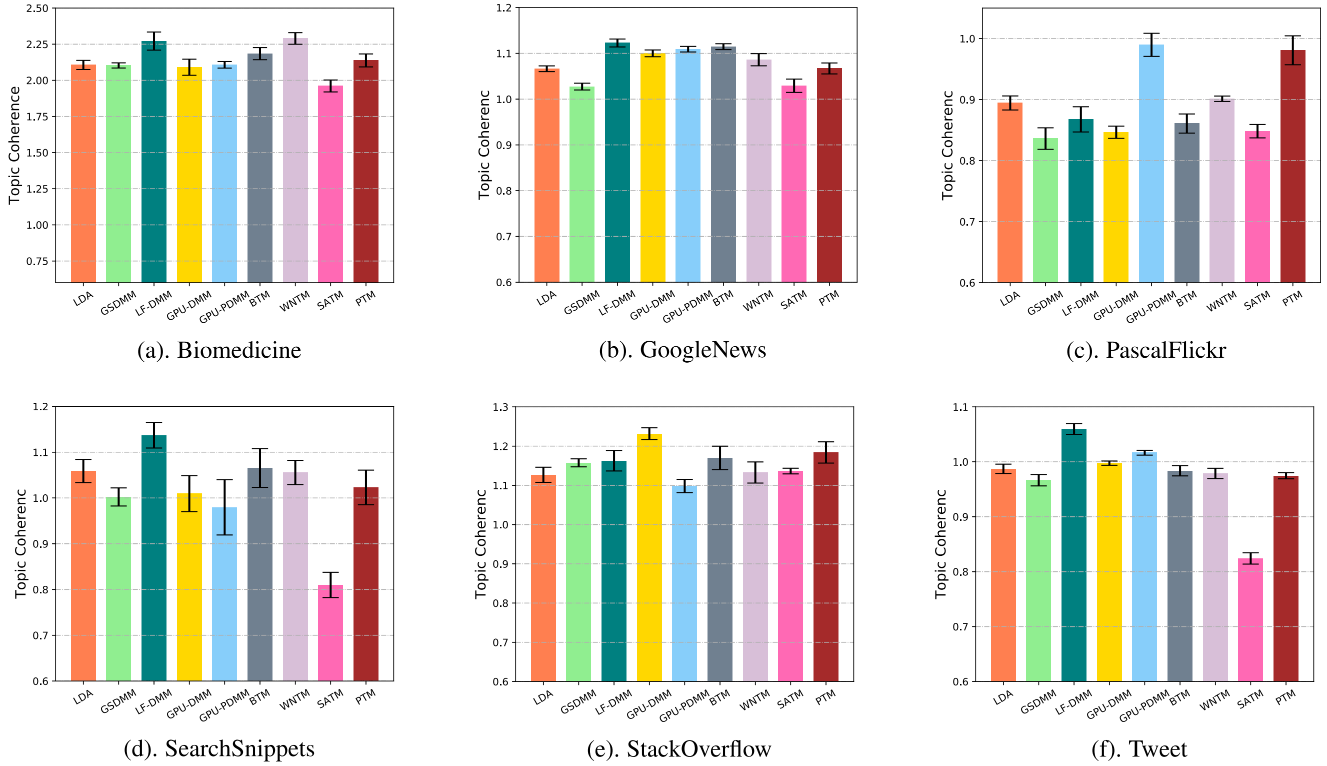
Topic coherence is used to evaluate the quality of topic-word distribution. Here, we only choose the top 10 words for each topic based on the word probability. The results are shown in Fig. 8. DMM based methods achieve the best performance on all datasets. LF-DMM has the best performance on four datasets (Biomedicine, GoogleNews, SearchSnippets, and Tweet), GPU-DMM has the best performance on StackOverflow, and GPU-PDMM achieves the best on PascalFlickr. It means that incorporating word embeddings into DMM can help to alleviate the sparseness. Two methods based on global word co-occurrences perform very well and achieve a similar result on each dataset, which indicates that the adequacy of global word co-occurrences can mitigate the sparsity of short texts. Similar to the above results using other metrics, self-aggregation based methods perform very poorly.

We also present the qualitative evaluation of latent topics. Here, we choose SearchSnippets dataset as an example, since it only contains eight topics that are Health, Politics-Society (politics), Engineering (engine.), Culture-Arts (culture), Sports, Computers, Business, and Education-Science (education). Table 6 shows the eight topics learned by the nine models. Each topic is visualized by the top ten words. Words that are noisy and lack of representativeness are highlighted in bold.

From Table 6, we observe that LF-DMM can achieve a similar conclusion with topic coherence, which can learn more coherent topics with fewer noisy and meaningless words. GPU-DMM and GPU-PDMM can not discriminate

TABLE 6
The Top Ten Words of Each Topic by Each Model on SearchSnippets Dataset

| LDA | | | | GSDMM | | | | LF-DMM | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Topic1 | Topic2 | Topic3 | Topic4 | Topic1 | Topic2 | Topic3 | Topic4 | Topic1 | Topic2 | Topic3 | Topic4 |
| (health) | (politics) | (engine.) | (culture) | (health) | (politics) | (engine.) | (culture) | (health) | (politics) | (engine.) | (culture) |
| health | **wikipedia** | car | music | health | political | car | movie | health | culture | motor | film |
| **information** | **encyclopedia** | engine | movie | **information** | culture | engine | **com** | cancer | party | engine | music |
| cancer | **wiki** | electrical | **com** | cancer | culture | **com** | art | disease | democratic | wheels | art |
| **gov** | **political** | **com** | **news** | gov | democracy | electrical | film | healthy | war | electronics | **com** |
| medical | **culture** | products | film | medical | **wikipedia** | **wikipedia** | fashion | medical | political | electric | fashion |
| **news** | **democracy** | digital | movies | **news** | party | system | **motor** | drug | democracy | cars | movie |
| research | **system** | home | yahoo | healthy | war | wheels | **wikipedia** | treatment | congress | models | books |
| disease | **party** | motor | art | disease | republic | **olympic** | books | physical | presidential | **phone** | arts |
| healthy | **republic** | energy | video | nutrition | **information** | digital | arts | food | communist | **graduation** | rock |
| nutrition | **philosophy** | calorie | arts | hiv | government | **trade** | movies | care | philosophy | **fashion** | band |
| Topic5 | Topic6 | Topic7 | Topic8 | Topic5 | Topic6 | Topic7 | Topic8 | Topic5 | Topic6 | Topic7 | Topic8 |
| (sport) | (computer) | (business) | (education) | (sport) | (computer) | (business) | (education) | (sport) | (computer) | (business) | (education) |
| com | computer | **gov** | edu | **news** | computer | business | research | sports | computer | business | research |
| **news** | business | **business** | research | **music** | software | market | edu | football | intel | financial | edu |
| sports | web | **information** | science | **com** | web | **news** | science | games | software | bank | graduate |
| football | software | **school** | theory | sports | programming | **information** | theory | **news** | device | economic | resources |
| games | **com** | **trade** | **journal** | football | **wikipedia** | stock | **information** | league | linux | trade | science |
| **amazon** | **news** | **edu** | theoretical | games | memory | gov | school | **com** | digital | **news** | university |
| game | **market** | **research** | physics | **movie** | **com** | **com** | university | hockey | network | market | school |
| soccer | **stock** | **home** | computer | game | intel | finance | journal | game | hardware | services | faculty |
| world | internet | **law** | **information** | **wikipedia** | internet | services | physics | soccer | web | law | center |
| tennis | programming | **economic** | university | tennis | data | **home** | **computer** | golf | computers | stock | national |
| **GPU-DMM** | | | | **GPU-PDMM** | | | | **BTM** | | | |
| Topic1 | Topic2 | Topic3 | Topic4 | Topic1 | Topic2 | Topic3 | Topic4 | Topic1 | Topic2 | Topic3 | Topic4 |
| (health) | (politics) | (engine.) | (culture) | (health) | (politics) | (engine.) | (culture) | (health) | (politics) | (engine.) | (culture) |
| health | political | **theory** | movie | health | **wikipedia** | **com** | culture | health | political | car | movie |
| **information** | culture | **theoretical** | music | gov | **encyclopedia** | **news** | art | **information** | **wikipedia** | engine | music |
| cancer | democracy | **physics** | **com** | cancer | **wiki** | **information** | **american** | gov | culture | **intel** | **com** |
| medical | **wikipedia** | **wikipedia** | film | medical | political | **home** | history | cancer | democracy | electrical | **amazon** |
| gov | party | **edu** | **news** | disease | system | **online** | **car** | medical | **encyclopedia** | **com** | film |
| **news** | system | **information** | movies | healthy | democracy | **world** | arts | **news** | party | digital | **news** |
| healthy | **information** | **science** | **wikipedia** | nutrition | party | **web** | **imdb** | research | **wiki** | motor | movies |
| disease | government | **research** | art | physical | government | **music** | museum | disease | system | wheels | books |
| nutrition | **news** | **amazon** | fashion | hiv | gov | **index** | **income** | healthy | government | products | art |
| hiv | gov | **com** | **amazon** | diet | war | **amazon** | literature | nutrition | war | automatic | video |
| Topic5 | Topic6 | Topic7 | Topic8 | Topic5 | Topic6 | Topic7 | Topic8 | Topic5 | Topic6 | Topic7 | Topic8 |
| (sport) | (computer) | (business) | (education) | (sport) | (computer) | (business) | (education) | (sport) | (computer) | (business) | (education) |
| sports | computer | business | research | **movie** | computer | business | research | sports | computer | business | edu |
| **news** | web | market | edu | sports | software | trade | edu | **news** | software | **news** | research |
| football | software | **news** | science | games | programming | management | science | football | web | market | science |
| **com** | programming | trade | school | football | systems | economic | theory | **com** | programming | **information** | theory |
| games | **com** | stock | journal | **yahoo** | memory | law | school | games | **com** | trade | **information** |
| soccer | wikipedia | **information** | university | game | **engine** | international | university | game | internet | stock | university |
| game | memory | **com** | computer | **video** | intel | gov | journal | soccer | memory | services | journal |
| **wikipedia** | intel | services | **information** | soccer | design | products | theoretical | match | data | **home** | school |
| tennis | linux | finance | department | **film** | electrical | jobs | physics | world | wikipedia | gov | theoretical |
| world | digital | **home** | graduate | **movies** | security | bank | department | tennis | linux | finance | physics |
| **WNTM** | | | | **SATM** | | | | **PTM** | | | |
| Topic1 | Topic2 | Topic3 | Topic4 | Topic1 | Topic2 | Topic3 | Topic4 | Topic1 | Topic2 | Topic3 | Topic4 |
| (health) | (politics) | (engine.) | (culture) | (health) | (politics) | (engine.) | (culture) | (health) | (politics) | (engine.) | (culture) |
| health | political | **music** | **movie** | health | **wikipedia** | **culture** | **movie** | health | **wikipedia** | **amazon** | music |
| **information** | **wikipedia** | **engine** | **com** | **information** | **research** | **amazon** | **news** | **information** | **encyclopedia** | **theory** | movie |
| cancer | culture | **car** | **amazon** | **news** | **trade** | **wikipedia** | **com** | cancer | **wiki** | **com** | **com** |
| hiv | **encyclopedia** | **rock** | **film** | research | **information** | **democracy** | **film** | medical | **political** | **theoretical** | **news** |
| healthy | system | **com** | **art** | gov | **wiki** | **books** | **wikipedia** | gov | **democracy** | **physics** | film |
| nutrition | democracy | **motor** | **books** | medical | **gov** | **com** | **movies** | **news** | **system** | **books** | movies |
| disease | party | **reviews** | **fashion** | **party** | **international** | **political** | **reviews** | disease | **party** | **car** | video |
| medical | government | **pop** | **online** | **home** | **journal** | **history** | **online** | healthy | **war** | **engine** | reviews |
| news | war | **band** | **movies** | disease | **programming** | **edu** | **digital** | nutrition | **government** | **models** | **intel** |
| diet | world | **wikipedia** | **video** | healthy | **business** | **encyclopedia** | **articles** | hiv | **house** | **electrical** | imdb |
| Topic5 | Topic6 | Topic7 | Topic8 | Topic5 | Topic6 | Topic7 | Topic8 | Topic5 | Topic6 | Topic7 | Topic8 |
| (sport) | (computer) | (business) | (education) | (sport) | (computer) | (business) | (education) | (sport) | (computer) | (business) | (education) |
| sports | computer | market | research | sports | system | business | edu | **news** | computer | business | research |
| football | web | business | science | games | **com** | **news** | science | sports | edu | **news** | science |
| **news** | programming | trade | edu | **com** | web | **com** | research | **com** | software | **information** | edu |
| games | intel | stock | school | **news** | computer | market | school | football | web | market | journal |
| **com** | memory | **news** | theory | football | **information** | yahoo | **information** | games | **school** | services | **art** |
| soccer | internet | **information** | **information** | game | **car** | stock | university | game | research | **com** | resources |
| game | systems | jobs | journal | world | wikipedia | internet | **program** | soccer | programming | trade | culture |
| tennis | **com** | finance | university | soccer | memory | services | **fashion** | **culture** | **university** | stock | **information** |
| match | data | **home** | theoretical | **online** | device | financial | **department** | world | **information** | **home** | directory |
| league | wikipedia | tax | physics | **wikipedia** | **engine** | **information** | **home** | tennis | systems | gov | library |



(a). Biomedicine

(b). GoogleNews

(c). PascalFlickr

(d). SearchSnippets

(e). StackOverflow

(f). Tweet

Fig. 8. Topic coherence of all models on six datasets.

the topic ’Engineering’. SATM remains the worst method in all short text topic models, which cannot discriminate three topics ’Engineering’, ’Politics-Society’ can ’Culture. Except LDA, PTM, WNTM, and SATM, other models can identify at least seven topics from all eight topics.

### 7.4 Influence of the Number of Iterations

In this subsection, we try to investigate the influence of the number of iterations to the performance of all models using NMI metric. Since all models have converged when the number of iterations reaches 2000, we vary the number of iterations from 2 to 2024.

The results are shown in Fig. 9. We can see that models based DMM can converge fast to the optimal solutions and almost get stable within 30 iterations. Models based global word co-occurrences get stable within 60 iterations. Models based self-aggregation has the slowest convergence speed and the lowest iterative performance.

### 7.5 Efficiency

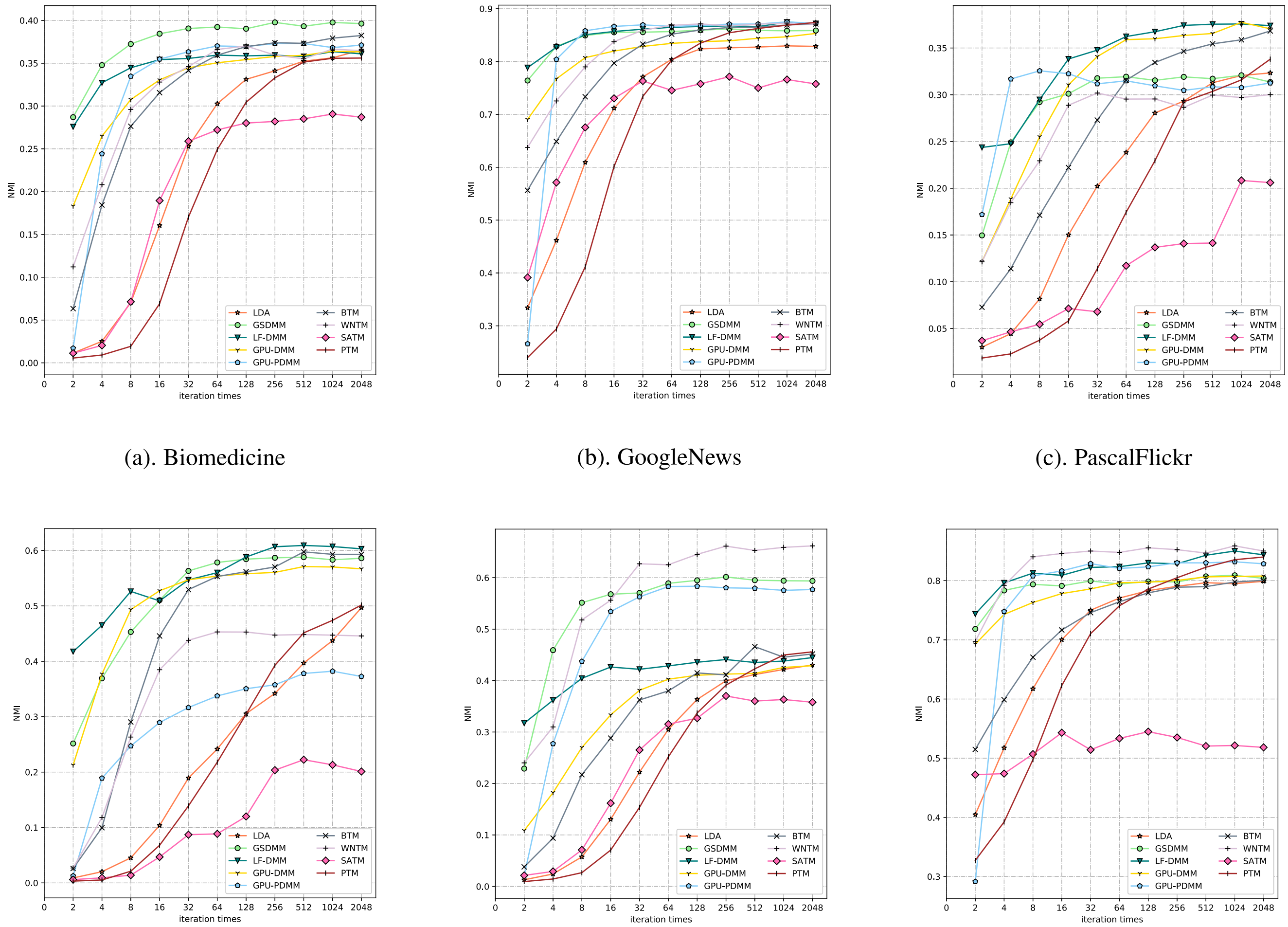
In this part, we compare the efficiency of various short text topic models. Here, we choose the largest dataset “Biomedicine” from all datasets to do the experiments.

The average runtime of the initiation and per iteration for each model are reported in Table 7. Among all models evaluated, LDA and DMM are the most efficient methods as expected. GPU-DMM is slightly slower than DMM and LDA, due to a similar Gibbs sampling process with GSDMM. LF-DMM and GPU-PDMM take much more time than GPU-DMM, because GPU-PDMM spends more time for the computational costs involved in sampling $\mathbf{Z}_d$ and LF-DMM need much time for optimizing the topic vectors. We can see that GPU-PDMM is the slowest modeling compared with other models.

Global word co-occurrences based methods are much slower than GSDMM, LDA and GPU-DMM and faster than the rest models. This is expected since they extend the number of words by extracting word co-occurrences. For self-aggregation based methods, the time is affected by the number of pseudo-documents. PTM is much faster than SATM but much slower than global word co-occurrences based methods. In addition, the models by incorporating word embeddings (GPU-DMM, LF-DMM, and GPU-PDMM) have the slowest time for the initiation due to the computational cost for the similarity between words.

### 7.6 Discussion

Based on the findings presented, we can state that strategies that exploit word embeddings (LF-DMM, GPU-DMM, and GPU-PDMM) are the most promising in short text topic modeling, which means that incorporating additional information can help to improve the performance of short text topic modeling. Two important cautions need to be noted. The first one is the computation cost. When computation cost is a major consideration, LF-DMM and GPU-PDMM are not a good choice. The second one is the strategies about incorporating word embeddings can significantly affect the performance of short text topic models. Global word co-occurrence based methods (BTM and WNTM) achieve better performance without any additional information than the self-aggregation based methods and LDA. BTM outperforms WNTM using classification and topic coherence. But, WNTM performs the best on several datasets for clustering. Self-aggregation based methods (SATM and PTM) performs



(a). Biomedicine (b). GoogleNews (c). PascalFlickr

(d). SearchSnippets (e). StackOverflow (f). Tweet

Fig. 9. NMI values with different number of iterations on every corpora.

slightly poorer than LDA on average. Moreover, SATM and PTM are affected by the number of pseudo-documents. Nonetheless, simpler methods (GSDMM and BTM) are the desired choice for short text topic modeling, with respect to both effectiveness and efficiency. For the clustering task, GSDMM achieves the best NMI value compared with other methods. BTM is not significantly affected by different datasets and tasks. Even GPU-PDMM achieves good results in most cases, it has a poor result on SearchSnippets dataset

TABLE 7
The Average Runtime of Initiation and Per Iteration of Each Model on Biomedicine (in Milliseconds)

| Model | Initiation time | Per iteration time |
|---|---|---|
| LDA | 77 | 41.50 |
| GSDMM | 46 | 48.15 |
| LF-DMM | 3329 | 2243.03 |
| GPU-DMM | 13610 | 53.83 |
| GPU-PDMM | 13037 | 9685.44 |
| BTM | 320 | 160.43 |
| WNTM | 192 | 220.12 |
| SATM | 41 | 2015.03 |
| PTM | 126 | 818.32 |

## 8 Conclusion and Future Work

The review of short text topic modeling (STTM) techniques covered three broad categories of methods: DMM based, global word co-occurrences based, and self-aggregation based. We studied the structure and properties preserved by various topic modeling algorithms and characterized the challenges faced by short text topic modeling techniques in general as well as each category of approaches. We presented various applications of STTM including content characterizing and recommendation, text classification, and event tracking. We provided an open-source Java library, named STTM, which is consisted of short text topic modeling approaches surveyed and evaluation tasks including classification, clustering, and topic coherence. Finally, we evaluated the surveyed approaches to these evaluation tasks using six publicly available real datasets and compared their strengths and weaknesses.

Short text topic modeling is an emerging field in machine learning, and there are many promising research directions:

1) Visualization: as shown in the survey, we display topics by listing the most frequent words of each topic (see Fig. 6). These new ways of labeling the topics may be more reasonable by either choosing

different words or displaying the chosen words differently [66], [67]. How to display a document using topic models is also a difficult problem? For each document, topic modeling provides useful information about the structure of the document. Binding with topic labels, this structure can help to identify the most interesting parts of the document.

2) Evaluation: Useful evaluation metrics for topic modeling algorithms have never been solved [10]. Topic coherence cannot distinguish the differences between topics. Besides, existing metrics only evaluate one part of topic modeling algorithms. One open direction for topic modeling is to develop new evaluation metrics that match how the methods are used.

3) Model checking: From the experimental results on this paper, each method has different performance on different datasets. When dealing with a new corpus or a new task, we cannot decide which topic modeling algorithms should I use. How can I decide which of the many modeling assumptions are suitable for my goals? The question of determining the interpretability of the models which Blei [10] labels as the model checking problem is among the most significant open issues facing topic modelers. One way of addressing the model checking problem is through interactive visualization supporting rapid experimentation for interpretive hypotheses [68]. New computational answers to these questions would be a significant contribution to topic modeling.

4) Deep learning: Deep learning (DL) techniques are capable of automatically learning low dimensional representations of things. Some researchers tried to model documents with layer-wise deep learning tools, including auto-encoders [69], restricted Boltzmann machine [70], document neural autoregressive distribution estimators [71] and deep Boltzmann machine [72]. Like topic models, the hidden layers in the deep networks provide the low-dimensional representation of documents. The main problem of deep learning is that it is hard to give each dimension of the generated distributed representations a reasonable interpretation. In recent years, some researchers combine the advantages of both topic models and neural networks. For example, deep learning techniques are used to conduct inference under the framework of a topic model [73]. The prior knowledge learned from the recurrent neural network is brought into biterms of BTM [74]. Joint deep learning with short text topic modeling serves as fruitful alternatives to explore in future research that would benefit from these types of advances that capture and exploit deep domain knowledge.

## Acknowledgments

This research was partially supported by the National Key Research and Development Program of China under grant 2016YFB1000900; the National Natural Science Foundation of China under Grants 61703362 and 91746209; the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China, under grant IRT17R32; and the Natural Science Foundation of Jiangsu Province of China under grant BK20170513.

## References

[1] T. Lin, W. Tian, Q. Mei, and C. Hong, "The dual-sparse topic model: Mining focused topics and focused terms in short text," in *Int. Conf. World Wide Web*, 2014, pp. 539–550.

[2] J. Qiang, P. Chen, W. Ding, T. Wang, F. Xie, and X. Wu, "Topic discovery from heterogeneous texts," in *Proc. IEEE 28th Int. Conf. Tools Artif. Intell.*, 2016, pp. 196–203.

[3] T. Shi, K. Kang, J. Choo, and C. K. Reddy, "Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations," in *Proc. World Wide Web Conf.*, 2018, pp. 1105–1114.

[4] X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 784–793.

[5] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2010, pp. 841–842.

[6] Z. Ma, A. Sun, Q. Yuan, and G. Cong, "Topic-driven reader comments summarization," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 265–274.

[7] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "TwitterRank: Finding topic-sensitive influential Twitterers," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 261–270.

[8] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 50–57.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[10] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[11] M. D. Hoffman, D. M. Blei, and F. Bach, "Online learning for latent Dirichlet allocation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 856–864.

[12] P. Xie and E. P. Xing, "Integrating document clustering and topic modeling," in *Proc. 29th Conf. Uncertainty Artif. Intell.*, 2013, pp. 694–703.

[13] P. Xie, D. Yang, and E. P. Xing, "Incorporating word correlation knowledge into topic modeling," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2015, pp. 725–734.

[14] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM: Topic modeling over short texts," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2928–2941, Dec. 2014.

[15] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang, "Transferring topical knowledge from auxiliary long texts for short text clustering," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 775–784.

[16] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & Web with hidden topics from large-scale data collections," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 91–100.

[17] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, "Improving LDA topic models for microblogs via tweet pooling and automatic labeling," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 889–892.

[18] Y. Wang, J. Liu, Y. Huang, and X. Feng, "Using hashtag graph-based topic model to connect semantically-related words without co-occurrence in microblogs," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1919–1933, Jul. 2016.

[19] J. Yin and J. Wang, "A Dirichlet multinomial mixture model-based approach for short text clustering," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 233–242.

[20] W. X. Zhao *et al.*, "Comparing Twitter and traditional media using topic models," in *Proc. Eur. Conf. Inf. Retrieval*, 2011, pp. 338–349.

[21] X. Quan, C. Kit, Y. Ge, and S. J. Pan, "Short and sparse text topic modeling via self-aggregation," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2270–2276.

[22] X. Yan, J. Guo, Y. Lan, J. Xu, and X. Cheng, "A probabilistic model for bursty topic discovery in microblogs," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 353–359.

[23] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, no. 2/3, pp. 103–134, 2000.

[24] G. Yu, R. Huang, and Z. Wang, "Document clustering via Dirichlet process mixture model with feature selection," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2010, pp. 763–772.

[25] R. Huang, G. Yu, Z. Wang, J. Zhang, and L. Shi, "Dirichlet process mixture model for document clustering with feature partition," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1748–1759, Aug. 2013.

[26] J. Qiang, Y. Li, Y. Yuan, and X. Wu, "Short text clustering based on Pitman-Yor process mixture model," *Appl. Intell.*, vol. 48, no. 7, pp. 1802–1812, 2018.

[27] J. Yin and J. Wang, "A text clustering algorithm using an online clustering scheme for initialization," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1995–2004.

[28] J. Yin, D. Chao, Z. Liu, W. Zhang, X. Yu, and J. Wang, "Model-based clustering of short text streams," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2634–2642.

[29] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *Trans. Assoc. Comput. Linguistics*, vol. 3, pp. 299–313, 2015.

[30] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Topic modeling for short texts with auxiliary word embeddings," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 165–174.

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[32] C. Li, Y. Duan, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Enhancing topic modeling for short texts with auxiliary word embeddings," *ACM Trans. Inf. Syst.*, vol. 36, no. 2, 2017, Art. no. 11.

[33] Y. Zuo, J. Zhao, and K. Xu, "Word network topic model: A simple but general solution for short and imbalanced texts," *Knowl. Inf. Syst.*, vol. 48, no. 2, pp. 379–398, 2016.

[34] W. Chen, J. Wang, Y. Zhang, H. Yan, and X. Li, "User based aggregation for Biterm topic model," in *Proc. 53rd Annu. Meet. Assoc. Comput. Linguistics 7th Int. Joint Conf. Nat. Lang. Process.*, 2015, pp. 489–494.

[35] W. Wang, H. Zhou, K. He, and J. E. Hopcroft, "Learning latent topics from the word co-occurrence network," in *Proc. Nat. Conf. Theor. Comput. Sci.*, 2017, pp. 18–30.

[36] Y. Zuo *et al.*, "Topic modeling of short texts: A pseudo-document view," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 2105–2114.

[37] J. Qiang, P. Chen, T. Wang, and X. Wu, "Topic modeling over short texts by incorporating word embeddings," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2017, pp. 363–374.

[38] P. V. Bicalho, M. Pita, G. Pedrosa, A. Lacerda, and G. L. Pappa, "A general framework to expand short text for topic modeling," *Inf. Sci.*, vol. 393, pp. 66–81, 2017.

[39] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," in *Proc. 1st Workshop Soc. Media Analytics*, 2010, pp. 80–88.

[40] A. K. McCallum, "Mallet: A machine learning for language toolkit". 2002. [Online]. Available: http://mallet.cs.umass.edu

[41] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 1–3, pp. 503–528, 1989.

[42] H. Mahmoud, *Pólya Urn Models*. London, U.K./Boca Raton, FL, USA: Chapman and Hall/CRC, 2008.

[43] M. Rosen-Zvi, T. L. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proc. 20th Conf. Uncertainty Artif. Intell.*, 2004, pp. 487–494.

[44] D. Ramage, S. Dumais, and D. Liebling, "Characterizing Microblogs with topic models," in *Proc. 4th Int. AAAI Conf. Weblogs Soc. Media*, 2010, pp. 130–137.

[45] Z. Dai, A. Sun, and X.-Y. Liu, "Crest: Cluster-based representation enrichment for short text classification," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2013, pp. 256–267.

[46] A. H. Razavi and D. Inkpen, "Text representation using multilevel latent Dirichlet allocation," in *Proc. Can. Conf. Artif. Intell.*, 2014, pp. 215–226.

[47] C. X. Lin, B. Zhao, Q. Mei, and J. Han, "PET: A statistical model for popular events tracking in social communities," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2010, pp. 929–938.

[48] C. C. Aggarwal and K. Subbian, "Event detection in social streams," in *Proc. SIAM Int. Conf. Data Mining*, 2012, pp. 624–635.

[49] A. Ritter *et al.*, "Open domain event extraction from Twitter," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 1104–1112.

[50] H. Yin *et al.*, "Discovering interpretable geo-social communities for user behavior prediction," in *Proc. IEEE 32nd Int. Conf. Data Eng.*, 2016, pp. 942–953.

[51] I. Guy, "Social recommender systems," in *Recommender Systems Handbook*. Berlin, Germany: Springer, 2015, pp. 511–543.

[52] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, Jul. 2014.

[53] O. Phelan, K. McCarthy, and B. Smyth, "Using twitter to recommend real-time topical news," in *Proc. 3rd ACM Conf. Recommender Syst.*, 2009, pp. 385–388.

[54] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi, "Short and tweet: Experiments on recommending content from information streams," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2010, pp. 1185–1194.

[55] H. Yin, B. Cui, L. Chen, Z. Hu, and X. Zhou, "Dynamic user modeling in social media systems," *ACM Trans. Inf. Syst.*, vol. 33, no. 3, pp. 1–44, 2015.

[56] M. Chen, X. Jin, and D. Shen, "Short text classification improved by learning multi-granularity topics," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1776–1781.

[57] D.-T. Vo and C.-Y. Ock, "Learning to classify short text from scientific documents using topic models with various types of knowledge," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1684–1698, 2015.

[58] J. H. Lau, N. Collier, and T. Baldwin, "On-line trend analysis with topic models: #twitter trends detection topic model online," in *Proc. Int. Conf. Comput. Linguistics*, 2012, pp. 1519–1534.

[59] H. Yin, B. Cui, H. Lu, Y. Huang, and J. Yao, "A unified model for stable and temporal topic detection from social media data," in *Proc. IEEE 29th Int. Conf. Data Eng.*, 2013, pp. 661–672.

[60] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2014, pp. 1532–1543.

[61] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using Amazon's mechanical turk," in *Proc. NAACL HLT Workshop Creating Speech Lang. Data Amazon's Mech. Turk Assoc. Comput. Linguistics*, 2010, pp. 139–147.

[62] C. Finegan-Dollak, R. Coke, R. Zhang, X. Ye, and D. Radev, "Effects of creativity and cluster tightness on short text clustering performance," in *Proc. 54th Annu. Meet. Assoc. Comput. Linguistics*, 2016, pp. 654–665.

[63] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2010, pp. 100–108.

[64] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2011, pp. 262–272.

[65] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 288–296.

[66] J. Chuang, C. D. Manning, and J. Heer, "Termite: Visualization techniques for assessing textual topic models," in *Proc. Int. Work. Conf. Adv. Vis. Interfaces*, 2012, pp. 74–77.

[67] C. Sievert and K. Shirley, "LDAvis: A method for visualizing and interpreting topics," in *Proc. Workshop Interactive Lang. Learn. Vis. Interfaces*, 2014, pp. 63–70.

[68] J. Murdock and C. Allen, "Visualization techniques for topic model checking," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 4284–4285.

[69] M. Ranzato and M. Szummer, "Semi-supervised learning of compact document representations with deep networks," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 792–799.

[70] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: An undirected topic model," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1607–1614.

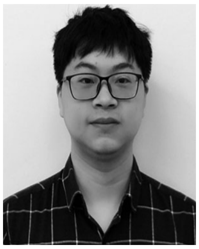
[71] H. Larochelle and S. Lauly, "A neural autoregressive topic model," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2708–2716.

[72] N. Srivastava, R. R. Salakhutdinov, and G. E. Hinton, "Modeling documents with deep Boltzmann machines," 2013, *arXiv:1309.6865*.

[73] Z. Cao, S. Li, Y. Liu, W. Li, and H. Ji, "A novel neural topic model and its supervised extension," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2210–2216.

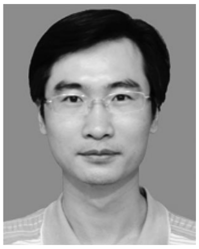
[74] H.-Y. Lu, L.-Y. Xie, N. Kang, C.-J. Wang, and J.-Y. Xie, "Don't forget the quantifiable relationship between words: Using recurrent neural network for short text topic discovery," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1192–1198.

**Jipeng Qiang** received the PhD degree in computer science and technology from the Hefei University of Technology, Hefei, China, in 2016. He is an assistant professor and the group leader of Computational Linguistics and Data Mining Group, Yanghou University. He was a PhD visiting student with the Artificial Intelligence Lab, University of Massachusetts Boston from 2014 to 2016. His research interests mainly include data mining and computational linguistics. He has received one grant from the National Natural Science Foundation of China, one grant from the Natural Science Foundation of Jiangsu Province of China, one grant from the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province of China. He has published more than 40 papers in data mining, artificial intelligence, and computational linguistics conferences and journals.



**Zhenyu Qian** received the BS degree in computer science from the Shanghai University of Electric Power, Shanghai, China. He is currently working toward the MS degree of computer science at the Yangzhou University, Yangzhou, China. His research interests include topic modeling and data mining.

**Yun Li** received the MS degree in computer science and technology from the Hefei University of Technology, Hefei, China, in 1991, and the PhD degree in control theory and control engineering from Shanghai University, Shanghai, China, in 2005. He is currently a professor with the School of Information Engineering, Yangzhou University, China. He has published more than 100 scientific papers. His research interests include data mining and cloud computing.



**Yunhao Yuan** received the MEng degree in computer science and technology from Yangzhou University, Yangzhou, China, in 2009, and the PhD degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2013. He is currently an associate professor with the School of information Engineering, Yangzhou University, China. His research interests include pattern recognition, data mining, and image processing.

**Xindong Wu** (Fellow, IEEE) received the BS and MS degrees in computer science from the Hefei University of Technology, Hefei, China, and the PhD degree in artificial intelligence from the University of Edinburgh, Edinburgh, Britain. He is a Yangtze River scholar with the School of Computer Science and Information Engineering, Hefei University of Technology, China, and the president of the Mininglamp Academy of Sciences, Mininglamp, Beijing, China, and a fellow of the AAAS. His research interests include data mining, big data analytics, knowledge-based systems, and Web information exploration. He is currently the steering committee chair of the IEEE International Conference on Data Mining (ICDM), the editor-in-chief of the *Knowledge and Information Systems* (KAIS, by Springer), and a series editor-in-chief of the Springer Book Series on Advanced Information and Knowledge Processing (AI&KP). He was the editor-in-chief of the *IEEE Transactions on Knowledge and Data Engineering* (TKDE, by the IEEE Computer Society) between 2005 and 2008. He served as program committee chair/co-chair for the 2003 IEEE International Conference on Data Mining, the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, and the 19th ACM Conference on Information and Knowledge Management.