World Wide Web (2020) 23:3099–3124
https://doi.org/10.1007/s11280-020-00823-w

Improving biterm topic model with word embeddings

Jiajia Huang[1] · Min Peng[2] · Pengwei Li[1] · Zhiwei Hu[3] · Chao Xu[1]

Received: 12 September 2019 / Revised: 29 April 2020 / Accepted: 4 May 2020 /
Published online: 8 September 2020

Abstract

As one of the fundamental information extraction methods, topic model has been widely used in text clustering, information recommendation and other text analysis tasks. Conventional topic models mainly utilize word co-occurrence information in texts for topic inference. However, it is usually hard to extract a group of words that are semantically coherent and have competent representation ability when the models applied into short texts. It is because the feature space of the short texts is too sparse to provide enough co-occurrence information for topic inference. The continuous development of word embeddings brings new representation of words and more effective measurement of word semantic similarity from concept perspective. In this study, we first mine word co-occurrence patterns (i.e., biterms) from short text corpus and then calculate biterm frequency and semantic similarity between its two words. The result shows that a biterm with higher frequency or semantic similarity usually has more similar words in the corpus. Based on the result, we develop a novel probabilistic topic model, named Noise Biterm Topic Model with Word Embeddings (NBTMWE). NBTMWE extends the Biterm Topic Model (BTM) by introducing a noise topic with prior knowledge of frequency and semantic similarity of biterm. NBTMWE shows the following advantages compared with BTM: (1) It can distinguish meaningful latent topics from a noise topic which consists of some common-used words that appear in many texts of the dataset; (2) It can promote a biterm's semantically related words to the same topic during the sampling process via generalized *Pólya* Urn (GPU) model. Using auxiliary word embeddings trained from a large scale of corpus, we report the results testing on two short text datasets (i.e., Sina Weibo and Web Snippets). Quantitatively, NBTMWE outperforms the state-of-the-art models in terms of coherence, topic word similarity and classification accuracy. Qualitatively, each of the topics generated by NBTMWE contains more semantically similar words and shows superior intelligibility.

Keywords Topic model · Word embeddings · Short texts · Noise biterm · BTM

✉ Pengwei Li
pwli@nau.edu.cn

Extended author information available on the last page of the article.

1 Introduction

With the development of social media, short texts have become popular information carriers on the Internet. The texts include tweets, questions in Q&A community, labels of images or videos, news titles and comments and so on. Discovering knowledge hidden in large scale of short texts has become a challenge and promising research issue, which is embodied as various tasks, such as topic extraction [8, 37, 38], emerging event detection [12, 26], comments summarization [23, 34], conversation generation [22], sentiment analysis [1, 9] and so on. Topic model, as a fundamental task among them, aims at discovering a group of latent topics from volumes of unlabeled texts.

Probabilistic topic models, such as Latent Dirichlet Allocation (LDA) [2] and its improvement versions [3, 4], have been widely used for topics mining in the past several years. In these models, a topic is represented as a multinomial distribution over words that can be inferred by Gibbs sampling or variational inference from the word co-occurrence information provided by the texts. However, due to the limited length of short texts, they contain much less co-occur words compared with that in normal texts. Thus, it results in the sparsity problem and makes the topics generated by these models be incoherent and elusive.

To alleviate the sparsity problem of implementing topic model on short texts, several strategies are proposed based on latent Dirichlet process. One strategy, namely Biterm Topic Model (BTM) [35], utilizes co-occurred word pairs (i.e., biterms) mined from the corpus for topic inference. In this model, biterms are mined from a short text corpus first and BTM is then implemented on the biterm collection directly. Another strategy, such as Dirichlet Multinomial Mixture (DMM) [38], restricts the text distribution over latent topics, namely, it assumes that each text covers a single topic. The third strategy is to aggregate similar short texts into a pseudo-document before implementing standard topic model. Here, hashtags, users, locations or timestamps are employed as the aggregation criterion [17, 28, 33]. Experimental results reported that these strategies are able to improve topic coherence compared with LDA [28, 35, 38].

All of the improvement strategies presented above only utilize the internal information of corpus while ignoring the external knowledge, such as the semantic similarity between two words. Word embeddings [18] are widely studied in recent years, aiming at training a novel semantic representation of word from large scale of documents (e.g., Wikipedia). Word embeddings have also been applied into topic model to tackle the data sparsity issue [6, 8, 16, 21]. For example, *Li* et al. proposed a generalized-*Pólya*-Urn-based Dirichlet Multinomial Mixture (GPUDMM) topic model [16], which combined the word embeddings with DMM model for topic inference. In these models, word semantic similarity based on word embeddings is exploited to improve topic quality.

In this study, we assert that both biterm frequency and similarity between its both words are important for topic inference. Word co-occurrence patterns can tell us which word pair co-occurs frequently in the corpus, while word pair similarity can tell us whether both words in the biterm are really semantically related or a casually collaboration occurring in this corpus. With these two metrics, it is reasonable to presume that there are some noise biterms in the biterm collection that cannot be assigned to any meaningful topics. In this sense, these noise biterms should be aggregated into a cluster that can be distinguished from the meaningful topics. Thus, a noise topic is introduced into the topic model to represent the cluster. Furthermore, the noise probability of each biterm can be estimated by the external semantic knowledge and the internal co-occurrence knowledge. That is to say, a biterm with lower values of frequency and semantic similarity has higher probability of being regarded as a noise biterm.

In this study, we propose a novel topic model for short texts clustering, named NBTMWE (Noise Biterm Topic Model with Word Embeddings), which is designed to alleviate the data sparsity problem by exploiting both the semantic similarity and frequency of biterm during the sampling process. As shown in Figure 1, the proposed NBTMWE model extends the Biterm Topic Model (BTM) by introducing a noise topic and employs the word embeddings pre-trained from a large scale of public corpus for topic inference. In detail, firstly, biterm frequency and similarity between its two words are utilized to evaluate the biterm's noise probability. Secondly, a biterm is sampled as a noise biterm or as a normal one depends on the Bernoulli distribution of its noise probability. Finally, in the sampling process, generalized *Pólya* Urn model (GPU) [10] is employed for topic promotion. That is to say, the semantically related words of a normal biterm are promoted to the same topic when a meaningful topic is assigned to the biterm. This promotion strategy is designed based on our statistical result that a biterm with high frequency or high semantic similarity usually has more highly related words.

Experimental results tested on two corpuses show that the proposed NBTMWE model is able to discover topics with higher topic word coherence and classification accuracy than existing state-of-the-art alternatives. The main contributions of this paper are summarized as follows:

– We make statistics from word pair co-occurring frequency, word pair similarity and word-biterm similarity with using real-world short texts datasets, including *Sina Weibo* and *Web Snippets*. The statistical results reveal that there is no significant correlation between biterm frequency and its similarity, while the average similarity between biterm and each of its semantically related words shows significant correlation with the

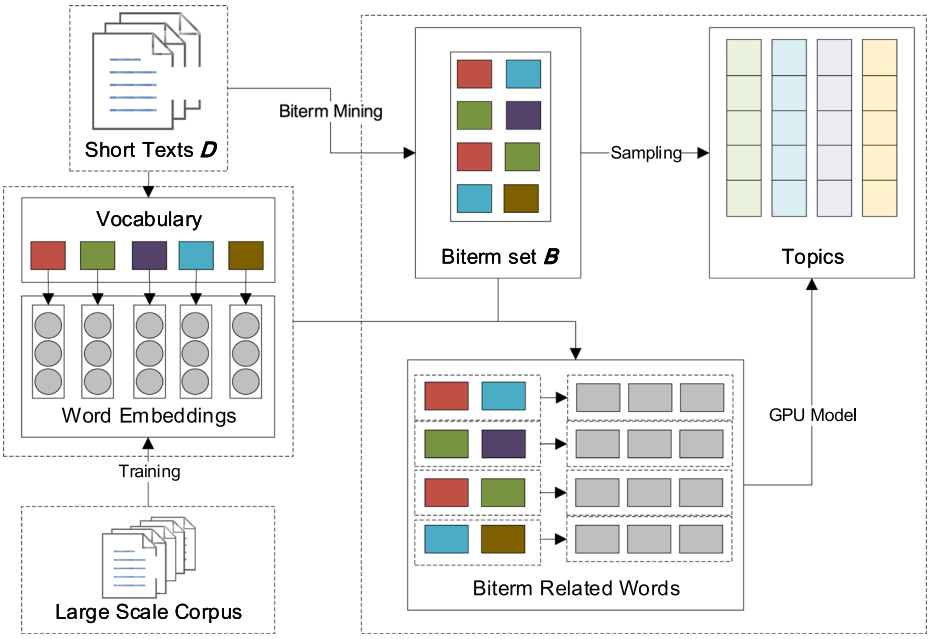


Figure 1 Architecture of NBTMWE

biterm's frequency and similarity respectively. This result provides a fundamental base for designing the topic promotion strategy.
– We propose a novel word-embeddings-based probabilistic topic model from biterm inference perspective. The key idea of our approach is to estimate the noise probability of biterm as the prior knowledge and then develop a noise topic from the traditional BTM based on the probability. This model has advantage of extracting both coherent and intelligible topics.
– Besides existing public testing corpus, namely *Web Snippets*, we also build another short text corpus (called *Sina Weibo*) that collected from Sina Weibo messages for experimental evaluation. Furthermore, a novel topic quality metric, namely *word-embeddings-based-similarity* [7] is also employed for topic coherence evaluation. The experimental results demonstrate that the proposed method outperforms the alternatives on both datasets in terms of topic coherence and text classification accuracy.

The rest of this paper is organized as follows. Section 2 briefly summarizes the related work of topic models used for short text clustering. Section 3 briefly introduces Biterm Topic Model and then presents the details of the proposed NBTMWE, including the statistics of biterms, noise probability of biterm, NBTMWE implementation and the model inference. We evaluate the method in Section 4 and draw a conclusion in Section 5.

2 Related works

The major challenge of short text clustering is the sparsity issue. In recent years, many studies focus on alleviating the issue by improving conventional topic models (e.g., LDA and DMM) from various aspects. As our model utilizes word co-occurrence knowledge and word semantics knowledge for topic inference, the following related studies are reviewed from these two aspects.

Topic Model using word co-occurrence knowledge Both LDA [2] and pLSA (Probabilistic Latent Semantic Analysis) [11] assume that a document can be regarded as a mixture distribution over K topics, but they perform poorly on short texts. Then the improved topic models, such as DMM [38], assume that a short text only covers one topic. Experimental results show that DMM achieves preferable performance on short texts due to the reasonable assumption [38].

In recent years, domain knowledge, especially word co-occurrence patterns mined from corpus is used to improve coherence and intelligibility of topical words. One improvement way is that words (including both *must-link* words and *cannot-link* words) mined from domain knowledge are employed to constraint the topical word generation process. That is to say, it makes the *must-link* words (i.e., the word co-occurrence patterns) be assigned to the same topic while the *cannot-link* words be assigned to different topics with high probabilities. Several topic models have been proposed based on this consideration, such as LTM (Lifelong Topic Model) [4] that uses *must-link* information and AMC (topic modeling with Automatically generated Must-links and Cannot-links) [3] that uses both links. In these models, link information is mined from domain corpus with frequent itemset mining. The other way of using word co-occurrence patterns is BTM [35], which uses biterm for topic inference. As the model is most relevant to our proposed model, we will introduce it briefly in Section 3.1. We choose BTM as our basic model is because BTM has shown a comparative performance on short text clustering based on existing experimental results [5,

16, 35]. In addition, *Jiang* et al. [13] proposed an extended model of BTM, namely biterm pseudo document topic model, which uses the word co-occurrence network to construct biterm pseudo texts before applying BTM.

Topic model using word semantics knowledge Word embeddings, one of the most typical ways of representing word vector, have been widely used in many NLP tasks, such as relation extraction [30], question answer [32] and topic model [16, 21, 25]. Due to the superior effectiveness in word semantic and syntactic similarity evaluation, word embeddings have also been introduced into topic models in recent years. Similar to our work, there are some existing works that use pre-trained word embeddings to promote similar words into the same topic. Among them, most relevant to ours work are GPUDMM [16], GPU-PDMM (Poisson-based GPU-DMM) [15] and LFDMM (Latent Feature Dirichlet Multinomial Mixture) [21]. In GPUDMM, DMM is employed as a basic topic model and a topic-word promotion strategy is designed based on GPU model to assign similar words into the same latent topic. In this method, when a word in a text is selected for promotion during sampling process, its semantically related words are also promoted to the same topic via the GPU model. Based on GPUDMM, *Li* et al. further proposed a Poisson-based GPU-DMM topic model, which introduced a Poisson distribution to determine the number of topics each text covers. In the sampling process, it first samples a number of topics as well as the topic assignments from the Poisson distribution for each text and then uses the topic-word promotion strategy for topical word assignment. In LFDMM, the topic-word promotion strategy is determined by a two-component mixture of a topic-word multinomial component and a latent feature component. Another similar topic models include Gaussian-LDA [6] and CRFTM (Conditional Random Field regularized Topic Model) [8]. Gaussian-LDA is designed without promotion selection strategy, but it uses multivariate Gaussian distribution with auxiliary word embeddings rather than Dirichlet-multinomial distribution to draw topical words. In CRFTM, it not only aggregates short texts into pseudo-documents, but also employs a conditional random field regularized model with word embeddings to promote semantically related words to the same topic.

Besides combining pre-trained word embeddings with topic model via a *two-step* method discussed above, some recent studies proposed unified models to learn word embeddings and latent topics [14, 29] at the same time because basic word embeddings methods perform poorly on polysemous words. For example, *Li* et al. [14] proposed a unified model that learns an embedding link function to connect the word embeddings and latent topic. *Shi* et al. [29] proposed a Skip-gram topical word embeddings method, which can obtain topic-specific word embeddings and term distributions over the latent topics at the same time.

Contrast to existing related works, our work combines both co-occurrence knowledge and semantic similarity knowledge of word pairs to improve topic model performance. To the best of our knowledge, NBTMWE is the first attempt of combining word embeddings with BTM to solve the sparsity problem of short texts.

3 Methodlogy

In this study, a model named NBTMWE is proposed for short texts clustering. Given a group of biterms mined from short texts, NBTMWE samples topics based on multinomial distribution that is similar to BTM. As shown in Figure 1, the auxiliary word embeddings employed in NBTMWE is pre-trained from large scale of documents, such as *Wikipedia*. Then the biterm similarity evaluated under embedding space cooperates with biterm frequency to

evaluate the noise probability of the biterm before topic inference process. Following, we first briefly introduce Biterm topic model and then present the details of the proposed NBTMWE.

3.1 Biterm topic model

Biterm topic model is a generative probabilistic model, which assumes that the latent topics over the whole text corpus can be learnt by modeling the generation of biterms in the corpus [5, 35] directly. Here, a biterm is defined as an unordered word-pair co-occurring in a text and the frequency of the biterm is the co-occurring times of the word-pair in the whole dataset. A major advantage of BTM is it can alleviate the sparsity problem at document level [35] by biterms.

Given a biterm collection $B = \{b_1, b_2, ..., b_{|B|}\}$ mined from a short text collection D with a vocabulary of size V. Each biterm b in B consists of a word pair w_1 and w_2 . BTM assumes that the corpus consists of a mixture of latent topics and the probability that a biterm drawn from a specific topic is captured by the chance that both words in the biterm drawn from the topic. Given the Dirichlet prior distribution parameters α and β, the generative process of the corpus in BTM can be described as follows.

1. for each topic z
2. draw a topic-specific word distribution $\phi_z \sim Dir(\beta)$;
3. draw a topic distribution $\theta \sim Dir(\alpha)$ for the whole corpus;
4. for each biterm b in B
5. draw a topic assignment $z \sim Multi(\theta)$;
6. draw two words $w_1, w_2 \sim Multi(\phi_z)$.

The hidden variables (i.e., z, θ, ϕ) in the generative process can be approximated by applying collapsed Gibbs sampling. Thus, in each iteration, a topic z is sampled via the conditional distribution as (1):

$$p(z_b = k|\mathbf{z}^{\neg b}, B, \alpha, \beta) \propto (m_k^{\neg b} + \alpha)\frac{(n_{w_i|k}^{\neg i} + \beta)(n_{w_j|k}^{\neg j} + \beta)}{(n_k^{\neg i} + V\beta)(n_k^{\neg i} + V\beta + 1)} \tag{1}$$

where $m_k^{\neg b}$ is the number of biterms assigned to topic k, and $n_{w_i|k}^{\neg i}$ is the number of times that word w_i is assigned to topic k, $n_k^{\neg i} = \sum_w n_{w_i|k}^{\neg i}$.

Correspondingly, the posterior probabilities of topic distribution and topic-word distribution can be calculated by point estimation as (2) and (3):

$$\phi_{w|z} = \frac{n_{w|z} + \beta}{\sum_w n_{w|z} + V\beta} \tag{2}$$

$$\boldsymbol{\theta}_z = \frac{m_z + \alpha}{\sum_z m_z + K\alpha} \tag{3}$$

3.2 Noise Biterm topic model with word embeddings

The goal of topic model is to discover latent topics from text corpus, where each topic is represented by a group of words as well as their the corresponding probabilities. The quality of a topic model is usually evaluated from coherence of the topic word and classification accuracy of the model. To obtain more coherent topics, some works employ word embeddings to promote semantically similar words to the same topic during the sampling process, such

as LFDMM [21] and GPUDMM [16], while other studies promote the word co-occurrence patterns to the same topic, such as LTM [3] and ACM [4].

All of the methods presented above are based on the premise that a document covers one topic or more. In LDA-based methods, such as Gaussian-LDA, LTM, ACM, a document is assumed as a mixture distribution over topics, while in DMM-based methods, a document is assumed to cover a single topic. Generally, the later methods show better performance on short texts. This assumption does alleviate the sparsity problem compared with LDA, but the assumption is too strong that may not suitable for the corpus that contains both long and short texts. For example, although tweets are regarded as short texts due to most tweets contain only a few terms, a part of tweets still contain more than 50 words that may express more than one topic. The longer tweets are usually more powerful in expressing opinion and propagating information and should get more attention. In this situation, it is hard to premise how many topics the text should cover.

Inspired by BTM, in this study, we also draw topics from biterms directly and assume that a biterm covers a single topic. Furthermore, we consider whether the frequency and semantic similarity of biterm can be exploited for topic quality improvement. For this purpose, we analyze the statistics of biterms first.

3.2.1 Statistics of biterms

In this study, two datasets named *Sina Weibo* and *Web Snippets* are employed for testing the performance of the proposed topic model and their basic statistics are shown in Table 1. Table 1 shows that although a dataset contains thousands of words after preprocess, the average length of the texts is less than 30, demonstrating the sparsity of the corpus from one aspect.

Next, we analyze the relationship between the frequency and similarity of a biterm.

Observation 1 For the whole biterm collection, there is no significant correlation between biterm frequency and its similarity.

Set $B = \{b\}$ as the biterm collection mined from texts, where t_b is the frequency of biterm b in corpus and s_b is the semantic similairty between its two words w_1 and w_2 that is estimated by the word embeddings trained from Wikipedia. In this study, we employ correlation coefficient to measure the degree of correlation between t_b and s_b by dividing their covariance by the square root of the product of their variances. It can vary from -1 (prefect negative correlation) through 0 (no correlation) to +1 (perfect positive correlation). Figure 2 shows the correlation coefficient between t_b and s_b of top-N biterms. Here, all biterms are sorted by their frequencies with N ranging from 3000 to 30000.

From Figure 2, we find that the correlation coefficient between frequency and similarity of biterm ranges in [0, 0.1] on both datasets, denoting that there is no significant correlation between the two variables. The reason is some biterms, even the high frequency biterms, don't show semantically relatedness between its two words. For example, in *Web Snippets* dataset, both frequencies of biterm (*information*, *home*) and biterm (*yahoo*, *directory*) are

Table 1 Statistics of two datasets

Dataset	♯Doc	♯Words	♯Biterms	♯Labels	Text length
Sina Weibo	9951	13421	273838	10	24.9
Web Snippets	12290	5590	362776	8	14.7

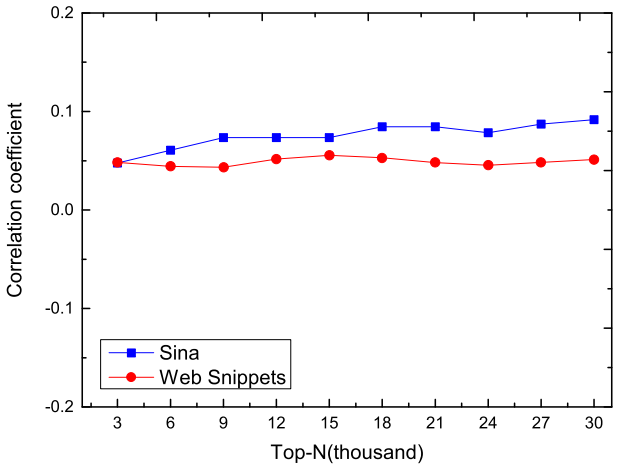
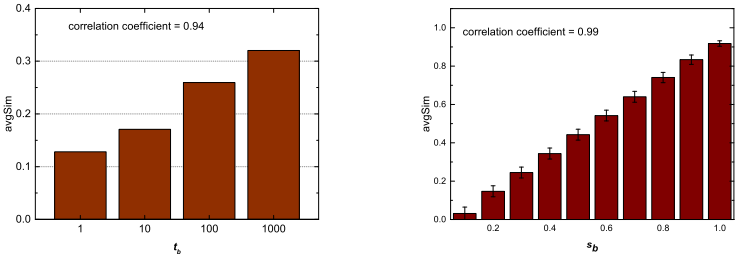


Figure 2 Correlation coefficient between biterm frequency and semantic similarity of top-N biterms with N ranging from 3000 to 30000 in two datasets

99 (very high), while both semantic similarities are 0.001 (very low). Furthermore, a biterm shows high frequency (e.g.,$t_{sport,com} = 100$, $t_{home,page} = 230$) may be because one word of the biterm is a common-used word (e.g., *information*, *home*, *page*) and co-occurs with different words frequently with low similarity (e.g., $t_{sport,com} = 0.23$, $t_{home,page} = 0.12$).

Then, we analyze the relationship between biterm and its semantically related words. Set $b = \{w_1, w_2\}$ as a biterm, $RW_b = \{w_b | s_{w_b,w_1} > \epsilon, s_{w_b,w_2} > \epsilon\}$ is the semantically related word set of biterm b, we want to investigate whether the semantic similarity $s_{w_b,b} = (s_{w_b,w_1} + s_{w_b,w_2})/2$ between b and $w_b \in RW_b$ is related to the frequency t_b or similarity s_b?

To investigate this problem, we select top-50 most similar words of each biterm to build its semantically related word set RW_b and then calculate the average value of $s_{w_b,b}$ for all biterms by grouping biterms with $\lg t_b$ and semantic similarities s_b respectively. Figure 3a and b show the statistical results of the two groups of average similarity testing on *Web Snippets* dataset.



(a) Grouping biterms by frequency

(b) Grouping biterms by semantic similarity

Figure 3 Average similarity between biterm and its related words by grouping biterms by frequency and similarity respectively. The correlation coefficient denotes the correlation between average similarity and $\lg t_b$ (or s_b)

Observation 2 There is significant correlation between biterm frequency and the average similarity with its related words.

Observation 3 There is significant correlation between biterm semantic similarity and the average similarity with its related words.

As shown in Figure 3a and b, testing results demonstrate that both correlation coefficients are very high, which denotes that when a biterm shows higher value of frequency or similarity, it has higher probability of owning more similar words. Based on this result, we consider the biterm with higher frequency or higher similarity is usually more important for topic inference than the lower one. This result will be used for designing our topic inference strategy.

3.2.2 Noise probability of biterm

Generally speaking, there are a large volume of biterms in a corpus, which show significant difference in frequency and semantic similarity. Does all of the biterms share equal opportunity in topic inference as assumed in BTM? Based on our statistical analysis results in Section 3.2.1, although there is no significant correlation between biterm similarity and frequency, the correlation between biterm frequency (or similarity) and the average similarity with its related words is significant. Therefore, in this study, we set some biterms as noise biterm and the noise probability of it is estimated by the metric of biterm frequency t_b and similarity s_b. The assumption is reasonable because when a biterm has a low value of t_b and s_b , the average similarity between it and its related words is usually also very low, indicating that the biterm may not be an important biterm in the biterm collection. Thus, the biterm should gain less attention in topic inference process.

Along this line, in this study, we first design a noise probability metric to estimate the importance of each biterm. Simply, the absolute importance of a biterm is set to $i_b = s_b t_b$. Since both words of a biterm might be a casual co-occurrence pair(e.g., (*yahoo*, *directory*)) or a frequent one (e.g.,(*www*, *http*), we decompose i_b into two parts: $i_b = i_{b0} + i_{b1}$, where i_{b0} is the normal count of biterm b in the dataset, while i_{b1} is the noise count of b. As i_{b0} and i_{b1} are not observed, we estimate their values based on all of the noise probabilities of b's related biterms as follows.

Suppose $RB_b = \{b_x = \{w_i, w_j\} | w_i \in b \text{ or } w_j \in b\}$ is the related biterm set of biterm b, namely, each biterm b_x (e.g., (*http*, *cn*)) in this set contains one word that is also contained by biterm b (e.g., (*www*, *http*)). We estimate the noise count $i_{b,1}$ of b by the mean of i_{b_x} of all biterms in set RB_b, i.e., $\widehat{i_{b,1}} = \frac{1}{|RB_b|} \sum_{b_x \in RB_b} i_{b_x}$. Consequently, we can obtain the normal count of biterm b as $\widehat{i_{b,0}} = (i_b - \widehat{i_{b,1}})_+$, where $(x)_+ = max(x, \delta)$, and δ is a very small positive number to avoid zero probability. In our experiments, δ is set to 0.001 after some preliminary experiments.

With $i_{b,0}$ and $i_{b,1}$ in hand, the noise probability of biterm b can be estimated as (4):

$$p_b = 1 - \frac{(i_b - \widehat{i_{b,1}})_+}{i_b} \tag{4}$$

We employ p_b as the probability of biterm b being regarded as a noise biterm in the dataset. The calculation of p_b implies that if a biterm has higher value of i_b compared with its related biterms, it is less likely to be regarded as a noise biterm.

3.2.3 Incorporating word similarity into noise biterm topic model

Before using auxiliary word embeddings to improve topic model performance, we first discuss how to draw a noise topic from the topic model. As discussed in Section 3.2.1, different biterms show different ability in similar words aggregation, where the biterm with higher frequency t_b or higher similarity s_b usually aggregates more similar words, and vise verse to the biterm with lower one. Furthermore, we propose a biterm noise probability metric that is estimated by t_b and s_b of a biterm as well as that of its related biterms. Inspired by the bursty-BTM proposed by *Yan* [36], which proposed a background topic when detecting bursty topics from tweet stream, we introduce a noise topic in this study based on the noise probabilities of biterms. As discussed above, when a biterm has higher value of noise probability p_b, it is more likely to be generated from a noise topic. In this study, Bernoulli distribution $I_b \sim Bern(p_b)$ is employed to denote whether biterm b is generated from a noise topic or not. $I_b = 0$ indicates b is generated from a noise topic, while $I_b = 1$ indicates b is generated from a meaningful topic. Finally, all topics, including both a noise topic and K meaningful topics, are set to multinomial distributions over the vocabulary (i.e., $\{\phi_z | z = 0, 1, ..., K\}$), where ϕ_0 denotes the word distribution over the noise topic and ϕ_z $(z = 1, 2, ..., K)$ denotes that of each meaningful topic.

Next, we discuss how to use the pre-trained word embeddings to improve similar words into the same meaningful topics.

When generalized $Pólya$ Urn (GPU) model [10] is exploited to explain the sampling process of topic model, it means when a word is selected by a given topic, a certain number of similar words are put back along with the original word to promote the topic. This idea indicates that word co-occurrence patterns can share the same topic. However, due to the short length of texts, the text corpus can't afford enough co-occurrence patterns for topic inference. In this study, when a biterm is selected by a topic, its semantically related words can also share the same topic. Thus, in the sampling process, sampling a biterm b with a meaningful topic z not only increases the count of both words $w_1, w_2 (\in b)$ to topic z, but also increases the count of the related words $w_b \in RW_b$ of biterm b to topic z.

GPU-based Promotion Strategy Along this line, we propose a topic promotion strategy in biterm sampling process, aiming at increasing the probability of the semantically related words and word co-occurrence patterns to the same latent meaningful topic.

As discussed above, different biterms show different ability in similar words aggregation, and the biterm with more semantically related words is expected for topic promotion. Therefore, we first propose a sample strategy to determine whether a biterm b is selected for topic promotion under a given meaningful latent topic z during the sampling process as (5)–(8):

$$S_b \sim Bern(\lambda_{b,z}) \tag{5}$$

$$\lambda_{b,z} = \frac{p(z|b)}{p_{max}(z'|b)} \tag{6}$$

$$p_{max}(z'|b) = max_k\, p(z = k|b) \tag{7}$$

$$p(z = k|b) = \frac{p(z = k)\,p(w_1|z = k)\,p(w_2|z = k)}{\sum_{i=0}^{K} p(z = i)\,p(w_1|z = i)\,p(w_2|z = i)} \tag{8}$$

where S_b indicates whether biterm b is selected for topic promotion. Observed from (5), (6), (7) and (8), if a biterm is highly relevant to latent topic z (i.e., S_b is more likely to be 1), then the promotion strategy is applied to both words of the biterm and its related words.

Next, we discuss the promotion degree for related words during the sampling process. In GPU model, not only the original word need to be put back, a certain number of similar words are put back at the same time. In our model, the biterm semantically related words are the similar words. When a biterm $b = \{w_1, w_2\}$ is selected for topic promotion, the promotion degree for w_1, w_2 and each of the word w_b in RW_b is set to given parameter $\mu \in [0, 1]$ as (9), which can achieve stable performance when it varies from 0 to 1.

$$A_{b,w} = \begin{cases} \mu, & if\ sim(w_1, w_b) > \varepsilon \quad and \quad sim(w_2, w_b) > \varepsilon \\ 1, & if\ w_b\ is\ w_1\ or\ w_2 \\ 0, & otherwise \end{cases} \tag{9}$$

where $sim(w_i, w_b)$ is the semantic similarity between w_i (i=1,2) and w_b evaluated by the cosine similarity of word embeddings. The minimize similarity threshold ε depends on similarity distribution of different corpus.

In summary, when a biterm b is sampled by a topic z, both words in the biterm as well as its related words will increase their count of degree $A_{b,w}$ under topic z. Thus, two words with higher similarity will have higher probability of belonging to the same topic z. After several iterations, the biterms with lower noise probabilities are more fasten on several meaningful topics while the biterms with higher noise probabilities are more likely to be assigned to the noise topic. Accordingly, the words of expressing the same topic have higher probabilities of aggregating into the same latent topic and have higher probabilities of representing this topic. Finally, the model generates a group of coherent topics of which words show significant intelligibility.

3.2.4 NBTMWE

Based on our discussion in Section 3.2.4, we conclude that the generative process of the biterms collection B in NBTMWE can be divided into two parts. The first part (line 4-5) is to determine whether samples a noise topic for each of the biterms in B via the Bernoulli distribution $I_b \sim Bern(p_b)$. The second part (line 6-15) is to draw topic words for current topic z. In this process, when a meaningful topic z is assigned to b and the topic promotion is employed, then the promotion strategy discussed above is adopted to promote b's related words into z. The generative process of NBTMWE is presented as Algorithm 1. Its graphical representation is shown in Figure 4, where $|B|$ denotes the number of biterms in B.

Algorithm 1 Generative Process of NBTMWE.

draw a meaningful topic distribution $\theta \sim Dir(\alpha)$;
for each topic $z \in [0, K]$ **do**
    draw a meaningful topic distribution $\phi_z \sim Dir(\beta)$;
end for
for each biterm $b \in B$ **do**
    draw $I_b \sim Bern(p_b)$;
    if $I_b = 0$ **then**
        draw two words $w_1, w_2 \sim Multi(\phi_0)$;
    else
        draw a meaningful topic $z \sim Multi(\theta)$;
        draw topic promotion $S_b \sim Bern(\lambda_{b,z})$;
        if $S_b = 0$ **then**
            draw two words $w_1, w_2 \sim Multi(\phi_z)$;
        else
            for word w_i in biterm related word set RW_b **do**
                draw word $w_i \sim Multi(\phi_z)$.
            end for
        end if
    end if
end for

3.2.5 Model inference

In NBTMWE, the GPU-based topic promotion strategy is integrated into the collapse Gibbs sampling process to infer latent topics. Accordingly, it needs to estimate the parameters alternatively by replacing the value of one variable by a value drawn from the posterior distribution of latent variables sequentially conditioned on the current values of all other

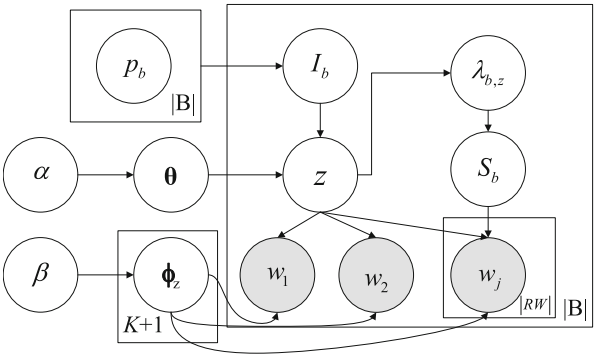


Figure 4 Graphical representation of NBTMWE

variables and the data. That is to say, the probability of drawing a topic for a biterm b depends on two latent variables, i.e.,z_b and I_b, which can be jointly estimated according the following conditional distribution.

$$p(I_b = 0|I^{\neg b}\mathbf{z}^{\neg b}, B, \alpha, \beta, p_b) = p_b \frac{(n^{\neg b}_{w_1|0} + \beta)(n^{\neg b}_{w_2|0} + \beta)}{(n^{\neg b}_{0} + V\beta)(n^{\neg b}_{0} + V\beta + 1)} \tag{10}$$

$$\begin{aligned} & p(I_b = k, z = k|I^{\neg b}\mathbf{z}^{\neg b}, B, \alpha, \beta, p_b) \\ = \ & (1 - p_b)\frac{(m^{\neg b}_{k} + \alpha)}{\sum_{k=1}^{K}(m^{\neg b}_{k} + \alpha)} \frac{(n^{\neg b}_{w_1|k} + \beta)(n^{\neg b}_{w_2|k} + \beta)}{(n^{\neg b}_{k} + V\beta)(n^{\neg b}_{k} + V\beta + 1)} \end{aligned} \tag{11}$$

Algorithm 2 NBTMWE.

Input:

Input Latent topic count K,prior parameters α and β, biterm collection B, noise probability vector of biterms $\mathbf{p} \in R^{1\times|B|}$

Output:

Output Topic distribution $\theta \in R^K$, topic word distribution $\phi \in R^{(K+1)\times V}$.

for each b in B **do**
  $z_b \leftarrow z \sim Multi(1/(K+1))$;
  NormalUpdateCounter(z_b, b, True);
  $S_b \leftarrow 0$
end for
for each iteration **do**
  UpdateBitermTopicProb(); //See (2), (3) and (8)
  for each b in B **do**
    $z \leftarrow z_b$;
    if $z = 0$ **then**
      NormalUpdateCounter(0, b, False);
    else
      RatioUpdateCounter(z, S_b, b, A_b, False);
    end if
    draw I_b, Z_b from (10) and (11)
    if I_b =0 **then**
      NormalUpdateCounter(0,b, True);
    else
      UpdatePromotionFlag(S_b); //see (5), (6) and (7);
      RatioUpdateCounter(z, S_b, b, A_b, True);
    end if
  end for
end for

Algorithm 3 NormalUpdateCounter(z_b, b, *increasement*).

Input:
Input topic assignment z_b, biterm b, *increasement* flag.
Output:
Output biterms assignment counter m_z, topic counter n_z, topic word counter n_z^w.

```
1: if increasement is True then
2:     m̃_z ← m̃_z + 1 ;
3:     for w in b do
4:         ñ_z ← ñ_z + 1;
5:         ñ_z^w ← ñ_z^w + 1;
6:     end for
7: else
8:     m̃_z ← m̃_z − 1 ;
9:     for w in b do
10:        ñ_z ← ñ_z − 1;
11:        ñ_z^w ← ñ_z^w − 1;
12:    end for
13: end if
```

Sampling process. Algorithm 2 presents the details of the collapse Gibbs sampling process of NBTMWE. Firstly, NBTMWE initializes the topic assignment for each biterms (Lines1-5). In each iteration of the sampling process, first calculate the biterm distribution conditioned on each of the topics (Lines 7) (including both noise topic and meaningful ones) based on (2), (3) and (8). If biterm b is generated from a noise topic (i.e., z= 0), then normally decrease corresponding counters (i.e., n_0, n_{0,w_1}, n_{0,w_2})(Lines 10-11), otherwise decrease the counter of each word in b's related word set based on the flag S_b (Lines 12-13). Next, we draw samples of the latent variables for each biterm according to (10) and (11) (Line 14). If sample a noise topic for biterm b, then normally increase the corresponding counters (Lines 16-17). If sample a new meaningful topic for biterm b, then the promotion flag of biterm b needs to update and the promotion strategy is applied depending on the flag (Lines 18-20). Namely, when $S_b = 1$, both words in b and the related words of b will increase its corresponding counters based on (9). Otherwise, simply update corresponding counts of both words in b. This iterative process continues until the predefined number of iterations is reached. At last, the parameters θ and ϕ can be estimated by (2) and (3).

Model Complexity. We now analysis the time complexity of NBTMWE and compare it with BTM. The time complexity of BTM is $O(K|B|)$ for one iteration, where K is the number of latent topics and $|B|$ is the size of biterms. Extended from BTM with introducing a noise topic, NBTMWE has the time complexity of $O((K+1)|B|+|B|\tau + (K+1)V)$, where τ depends on the average number of related words of a biterm, $(K+1)V$ is the time consumption used for calculating the word probability conditioned on topic $p(z=k|w)$ when updating biterm-topic probability distribution. In our experiment, the average value of τ is 2.39 and 8.46 implemented on *Sina Weibo* and *Web Snippets* respectively.

Algorithm 4 RatioUpdateCounter(z_b, S_b, b, A_b, *increasement*).

Input:
Input topic assignment z_b, biterm b, word semantic similarity matrix A, topic promotion flag S_b, *increasement* flag.

Output:
Output biterms assignment counter m_z, topic counter n_z, topic word counter n_z^w.

if *increasement* is True **then**
  $\widetilde{m}_z \leftarrow \widetilde{m}_z + 1$;
  if $S_b = 1$ **then**
    for w in b's related word set RW_b **do**
      $\widetilde{n}_z \leftarrow \widetilde{n}_z + A_{b,w}$;
      $\widetilde{n}_z^w \leftarrow \widetilde{n}_z^w + A_{b,w}$;
    end for
  end if
  for w in b **do**
    $\widetilde{n}_z \leftarrow \widetilde{n}_z + 1$;
    $\widetilde{n}_z^w \leftarrow \widetilde{n}_z^w + 1$;
  end for
else
  $\widetilde{m}_z \leftarrow \widetilde{m}_z - 1$;
  if $S_b = 1$ **then**
    for w in b's related word set RW_b **do**
      $\widetilde{n}_z \leftarrow \widetilde{n}_z - A_{b,w}$;
      $\widetilde{n}_z^w \leftarrow \widetilde{n}_z^w - A_{b,w}$;
    end for
  end if
  for w in b **do**
    $\widetilde{n}_z \leftarrow \widetilde{n}_z - 1$;
    $\widetilde{n}_z^w \leftarrow \widetilde{n}_z^w - 1$;
  end for
end if

4 Experiments

4.1 Experimental Setup

4.1.1 Dataset

Sina Weibo.[1] This dataset is a collection of 145894 weibo messages that discuss ten hot social issues. As the raw corpus contains some noise texts that may not relate to any of the labeled issues, we only select a subset of high-quality texts based on our previous work of high-quality text extraction [24]. In the extraction process, attributions such as review number, forward number, content quality and URL quality are utilized for text quality evaluation. Then EM algorithm is employed to measure the quality of each text in the raw corpus. Finally, top-10000 messages with the highest quality score are selected

[1]This dataset is available at https://github.com/Jenny-HJJ/NBTMWE/tree/Jenny-HJJ-master-dataset-sina

as the *Sina Weibo* dataset. The pre-processing of the dataset contains following steps: (1) convert traditional Chinese to simple Chinese via openCC;[2] (2) remove all stop words and the words with frequency less than 3; (3) only reserve the biterms with frequency higher than 3 (including 3) since millions of biterms are mined while most of them appear one or two times; (4) delete the extremely short text that contains less than 3 words. Finally, the corpus contains 9951 texts and 13421 words. Table 1 shows the statistics of the dataset, including corpus size, vocabulary size, biterm collection size, number of ground truth labels (labels), and average number of words per text.

Web Snippets.[3] This dataset is a collection of 12340 Web search snippets from 8 categories that has been used in a few studies [16, 27, 31]. In the pre-process, (1) covert letters to lowercase; (2) remove all of the non-alphabetic characters and stop words;[4] (3) remove the words with frequency lower than 3; (4) delete the extremely short text that contains less than 3 words. Finally, the corpus contains 12290 texts and 5590 words, as shown in Table 1.

It is worth noting that the average length of texts in Sina Weibo is higher than that in Web Snippets, and the former dataset contains more words.

Word Embeddings training In this study, 0.28 million Chinese Wikipedia articles[5] and 4.2 million English Wikipedia articles[6] are employed to pre-train 200-dimensional word embeddings via CBOW algorithm[7] with default parameters. A word's embedding is set as a zero vector if the word is not indexed by the external knowledge.

4.1.2 Baselines

We compare our proposed NBTMWE against the following topic models.

- **DMM** (Dirichlet Multinomial Mixture) [38] is a short text clustering model which assumes that each short document covers one topic.
- **BTM** (Biterm Topic Model) [35] learns topics from biterm collection, which assigns both words in a biterm to the same topic during the sampling process. This model works as our base model.
- **GPUDMM** [16] integrates word embeddings into DMM by promoting the semantically related words under the same topic via the GPU model. We use the implementation provided by the authors.[8] In addition, we also use the recommended settings that the amount of promotion μ for each semantically related word is set to 0.1 and 0.3 for *Sina Weibo* and *Web Snippets* respectively.
- **LFDMM** (Latent Feature model with DMM) [21] integrates word embeddings into DMM, in which the topic-word promotion strategy is determined by a two-component mixture of a topic-word Dirichlet multinomial component and a word embedding component. In this method, only the words with embeddings are reserved. We use

[2]The implementation of openCC is available at https://github.com/argolab/OpenCC

[3]This dataset is available at https://github.com/Jenny-HJJ/NBTMWE/tree/Jenny-HJJ-dataset-web-snippets

[4]Stop word list is downloaded from NLTK : http://www.nltk.org/

[5]The Chinese Wikipedia dataset is available at https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2

[6]The English Wikipedia dataset is available at https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-latest-pages-articles.xml.bz2

[7]The implementation of CBOW is available at https://code.google.com/p/word2vec

[8]Authors implementation of GPUDMM is available at https://github.com/NobodyWHU/GPUDMM

the implementation provided by the authors and use the recommended settings with $\lambda = 0.6$, $\alpha = 0.1$, $\beta = 0.01$ as in their paper.[9]

– **NBTM** (Noise Biterm Topic Model), a simplified version of the NBTMWE, which only introduces a noise topic with biterm noise probability metric but does not employ the topic promotion step during topic inference process.

Prior distribution parameters $\alpha = 50/K$ and $\beta = 0.01$ for all methods except LFDMM. We run Gibbs sampling for 500 iterations and reported the average result over 5 runs. We evaluate the performance of all methods in terms of topic coherence and text classification accuracy.

As for NBTMWE, it needs to set the parameter ε for biterm semantically related words selection. As the parameter depends on both testing dataset and the pre-trained word embeddings, we randomly sample some biterms from the dataset for quantile statistics. In detail, we randomly select 100 biterms and top-100 most related words of each biterm and then calculate the similarity between biterm and each of its related words. At last, sort all similarity values of all biterms in descending order and set ε to the upper quartile of the similarity list. In this study, ε is set to 0.4 and 0.3 in *Sina Weibo* and *Web Snippets* datasets respectively. The topic model[10] is implemented in Java and is carried out on a Window 10 server in Inter Core i5 3.0GH and 8G memory.

4.2 Noise topic and noise biterm evaluation

In this study, a noise topic is introduced to aggregate the noise biterms into a cluster. To observe the word distribution over the noise topic, Table 2 shows the top-10 most relevant topic words of the noise topic interfered from *Web Snippets* with K ranging in {40, 60, 80}. From Table 2, we find that the top-10 topic words basically don't vary with the number of topics. It is because these words appear frequently in the dataset while they are poor in distinguishing and expressing different topics, which can be verified by their frequencies and chi statistic scores.[11] In fact, the top-10 topic words are covered by the top-40 most frequency words of the vocabulary and most of then have very low values of chi-score. The results denote that the proposed model is able to aggregate the non-sense words into a noise topic although these words appear frequently in the dataset.

Furthermore, to investigate which biterms are most likely to be regarded as noise biterms in the sampling process, we use the relevant biterms of noise topic in *Web Snippets* with K=60 for testing. Namely, calculate the biterm-topic probability distribution (i.e., $p(z|b)$ (8) and select the relevant biterm that satisfying $arg_k max\ p(z|b) = 0$. Table 3 presents the top-5 most relevant biterms after all of the relevant biterms sorted in descending order by $p(z = 0|b)$. These biterms have high probabilities of belonging to the noise topic while have low values in terms of biterm frequency t_b and similarity s_b. The results verify that they are more likely to be noise biterms that occasionally co-occur in several texts.

Both the most relevant topic words presented in Table 2 and the most relevant biterms presented in Table 3 indicate that it is reasonable to introduce a noise topic into the topic model. Furthermore, the proposed noise biterm estimation strategy and noise topic generation strategy of NBTMWE do able to identify these non-sense biterms, aggregate them into a noise topic and also shows stable performance with varying K.

[9] Authors implementation of LFTM is available at https://github.com/datquocnguyen/LFTM

[10] The implementation of NBTMWE is available at https://github.com/Jenny-HJJ/NBTMWE

[11] The chi-scores of each word in the vocabulary of the dataset is calculated by sklearn.chi2.

Table 2 Top-10 topic words of noise topic in *Web Snippets* with K ranging in {40, 60, 80}

♯Topics	Topic words
K=40	information, news, page, research, wikipedia, index, web, home, world, online
K=60	information, news, page, research, wikipedia, index, web, home, online, world
K=80	information, news, page, research, wikipedia, index, web, world, home, online

4.3 Topic coherence evaluation

Coherence To investigate the quality of topic discovered by different methods, coherence score proposed by *Mimno* et al.[19] is employed for evaluation. Given a topic z and its top-N most relevant words $z_k = \{w^{(1)}, w^{(2)}, ..., w^{(N)}\}$, the coherence score of topic z is defined as (12):

$$C(z_k) = \sum_{t=2}^{T} \sum_{j=1}^{t} log(f_{i,j} + 1)/f_j \tag{12}$$

where f_j is the frequency of word $w^{(j)}$ appearing in the dataset, $f_{i,j}$ is frequency of $w^{(i)}$ and $w^{(j)}$ co-occurring in the dataset. Higher of $C(z_k)$ means more coherent among the topic words. In this study, we calculate the average coherence score of all latent topics as the overall quality of the model.

Figures 5 and 6 report the overall coherence of the topics discovered by different methods testing on two datasets with $N = \{5, 10, 20\}$ and $K = \{40, 60, 80\}$, respectively. On the whole, NBTMWE achieves the best performance on both datasets and the improvement over the base method is significant. BTM and GPUDMM achieve the second best performance on *Sina Weibo* and *Web Snippets* respectively. This metric evaluates the topic coherence based on co-occurrence words in the testing dataset. The outperformance of our proposed model demonstrates that the model shows superior ability in aggregating the co-occurrence words into the same topic.

WESim *Fang* et al proposed another topic coherence metric based on word embeddings [7], namely WESim. The result [7] tested on Twitter dataset shows that WESim can capture the topic coherence with more robust and efficient performance than PMI [20] . In this study, we also employ this metric for topic coherence evaluation. Given a topic z and its top-N most relevant words $z_k = \{w^{(1)}, w^{(2)}, ..., w^{(N)}\}$, the WESim score of topic z is defined as (13):

$$WESim(z_k) = \frac{2}{N(N-1)} \sum_{j=2}^{N} \sum_{i=1}^{j} sim(w_i, w_j) \tag{13}$$

Table 3 Top-5 most relevant biterms of noise topic, along with their frequencies and word similarity values

Biterm words	$p(z = 0/b)$	t_b	s_b
personal, contact	0.999	3	0.054
class, contact	0.999	3	0.077
post, personal	0.999	1	0.083
background, contact	0.999	1	0.003
role, personal	0.999	1	0.033

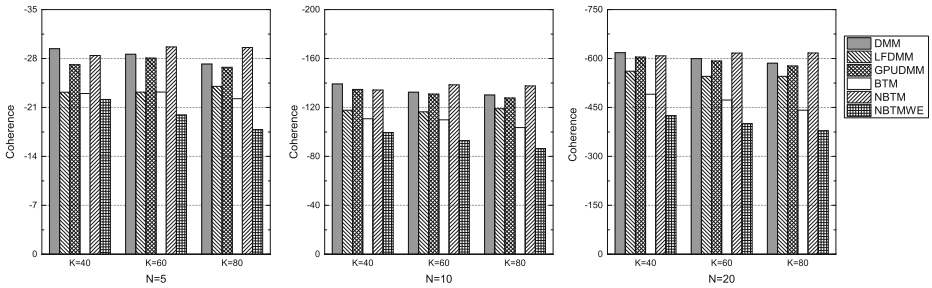


Figure 5 Topic Coherence evaluated on *Sina Weibo* Dataset

where $sim(w_i, w_j)$ denotes the semantic similarity between topic word w_i and w_j. The similarity is calculated based on word embeddings pre-trained by Wikipedia corpus and ranges in [-1,1]. Higher of WESim score means more semantic similar among the topic words. In this study, we calculate the average WESim score of all latent topics as the overall quality of the model.

Figures 7 and 8 compare the overall WESim score of topics discovered by different methods testing on two datasets with $N = \{5, 10, 20\}$ and $K = \{40, 60, 80\}$, respectively. Similar to Coherence metric, NBTMWE also achieves the best performance and the improvement is also significant. NBTM and GPUDMM achieve the second best performance on *Sina Weibo* and *Web Snippets* respectively. This metric evaluates the topic coherence based on topic word similarity that the word embeddings are pre-trained by a large scale of external corpus. In addition, we also used the PMI [20] metric for topic word coherence evaluation. Due to the space limitation, the results are not presented in the paper. Basically, the overall results are consistent with the WESim results, namely NBTMWE achieves the best performance while NBTM achieves the second best performance on both datasets. The outperformance of our proposed model demonstrates that the topic words generated by the model show better semantically coherence and superior ability in expressing topics.

At last, comprehensively analyzing the performance of different methods in terms of the metrics, two possible reasons can be used to explain the outperformance of our proposed model. On one hand, a noise topic is introduced into our model that is able to distinguish the non-sense words from the real topic words, which improves the topic coherence finally. On the other hand, word similarity evaluated by their embeddings is employed to promote the semantically related words into the same topic during the sampling process. Thus, each

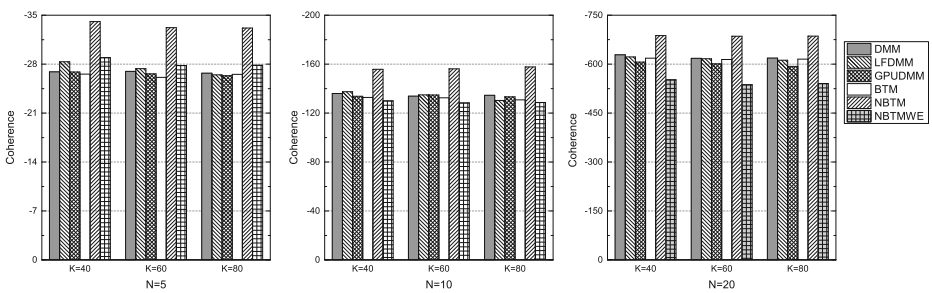


Figure 6 Topic Coherence evaluated on *Web Snippets* Dataset

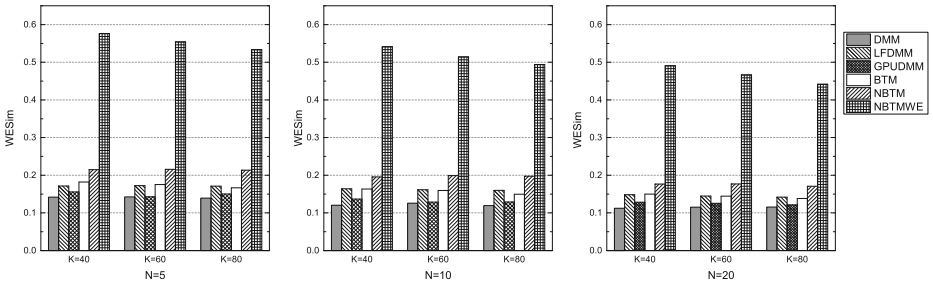


Figure 7 Topic WESim evaluated on *Sina Weibo* Dataset

of the inferred topics consists of similar topic words, which finally shows higher values in terms of topic word coherence metrics.

4.4 Qualitative evaluation of topic word

In practice, a better topic model should discover more readable and coherent topic word. For qualitative evaluation, we compare the top-10 words discovered by NBTMWE and GPUDMM under K=60 respectively. First, since each of the texts in the dataset has a category tag, we select the top-10 keywords based on their tf-idf score from the texts of each category. Then pick out the latent topic that covers the most number of the keywords for presentation. Due to space limitation, we only select the latent topics that most related to *computers*, *politics-society* and *sports* in *Web Snippets* dataset for demonstration. Topic words that appear in the keyword sets are highlighted in boldface. The overall results are shown in Table 4.

From Table 4, we find that the discovered topic words are generally semantically consistent and can express the connotation of corresponding category tag. Furthermore, the similarity of topic word generated by NBTMWE is much higher than that by GPUDMM. In addition, some words in the top-10 keywords extracted from each category do not have category distinguishing ability, such as *wikipedia*, *com* and *news*. This so-called non-sense words are also listed as the topic word in GPUDMM but not in NBTMWE. Combining the noise topic word presented in Table 2, we find that these words are aggregated into the noise topic in NBTMWE.

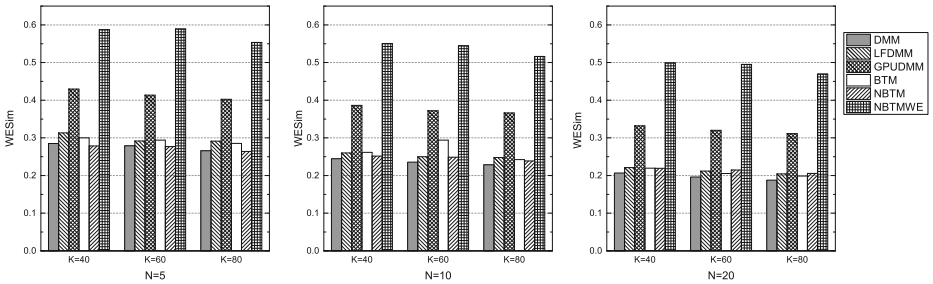


Figure 8 Topic WESim evaluated on *Web Snippets* Dataset

Table 4 Representative words of each latent topic discovered by NBTMWE and GPUDMM

Category tag	Top-10 Topic words				Keywords	
	NBTMWE		GPUDMM			
	Topic word	WESim	Topic word	WESim	Words	WESim
computers	computing, technology, **software**, **computer**, automation, ibm, hardware, computation, interface	0.584	**programming**, **web**, system, **computer**, program, **software**, xml, language, server, **wikipedia**	0.253	computer, web, software, programming, wikipedia, memory, com, internet, intel, information	0.316
politics-society	**democracy**, **political**, socialism, ideology, socialist, politics, communism, republicanism, pluralism, reform	0.548	**political**, **democracy**, **system**, **government**, election, **party**, **republic**, president, parliamentary, senate	0.361	political, democracy, party, system, wikipedia, government, war, news, republic, gov	0.224
sports	basketball, **football**, athletics, **soccer**, **sports**, **team**, hockey, lacrosse, baseball, tournament	0.579	**news**, **sports**, **football**, **team**, espn, league, **com**, **game**, **match**, nfl	0.393	sports, game, football, news, com, soccer, tennis, wikipedia, team, match	0.246

4.5 Classification accuracy evaluation

To investigate the performance of topic model on text classification, we represent each short text with its topic distribution $p(z|d)$. In this study, we use linear kernel Support Vector Machines (SVM) classifier in sklearn[12] with default parameters for short text classification. The linear kernel SVM is a conventional tool of evaluating classification accuracy in topic models, such as in BTM [35], LFTM [21] and GPUDMM [16]. The classification accuracy is computed through 5-fold cross validation on both datasets. As each of the texts in both datasets belongs to one of the labeled categories, we use the text labels to evaluate the classification accuracy.

As DMM and GPUDMM assume that each short text covers one topic, we use summation over words (SW) representations for text inference as (14), which has been proved of achieving better performance than Naive Bayes [16]:

$$p(z = k|d) \propto \sum_{w \in d} p(z = k|w)p(w|d) \tag{14}$$

where $p(w|d)$ is estimated via the relative frequency of w appearing in short text d.

[12]The implementation is available at http://scikit-learn.org/

In BTM, a variant of SW by replacing w with biterm b is used for document classification as (15) [35]:

$$p(z = k|d) \propto \sum_{b \in d} p(z|b)\, p(b|d) \tag{15}$$

where $p(z|b)$ is discussed in (8), $p(b|d) = n_d / \Sigma_b n_d(b)$, $n_d(b)$ is the frequency of biterm b appearing in short text d.

In this study, (14) is used for calculating $p(z|d)$ in DMM, LFDMM and GPUDMM, according to the presented best reported result [16]. (15) is used to calculate $p(z|d)$ as discussed in BTM [35]. In our proposed method, as a noise topic is extended from BTM, (15) is not appropriate for calculating $p(z|d)$ in NBTM and NBTMWE anymore. Here, we still use (14) to evaluate document classification result. The average classification accuracy of each method tested on both datasets are reported in Table 5 with $K = \{40, 60, 80\}$.

From the table, we make following observations.

Our proposed method achieves overall best performance than the alternative methods on both datasets. BTM and GPUDMM achieve the second best performance on *Sina Weibo* and *Web Snippets* respectively. Particularly, NBTMWE outperforms NBTM significantly on both datasets, which validates that incorporating word semantic similarity knowledge into topic inference process contributes much benefit for topic model.

Second, comparing the results testing on two different datasets, we find DMM-based methods, including DMM, LFDMM and GPUDMM, achieve better results on Web Snippets, while BTM based methods, including BTM and NBTMWE, achieve better results on *Sina Weibo*. It is because more texts in *Sina Weibo* may express more than one topic, which is reflected as the average text length of *Sina Weibo* is higher than that of *Web Snippets* and some texts in *Sina Weibo* incline to discuss several hot issues in one text. Therefore, in this situation, it is more appropriate to assume that a biterm covers one topic in BTM-based methods rather than a text covers a topic.

Table 5 Average classification accuracy on both datasets, with $K = \{40, 60, 80\}$ .The best results are highlighted in boldface on each dataset

Dataset	Methods	K=40	K=60	K=80
Sina Weibo	DMM	0.844	0.884	0.885
	LFDMM	0.820	0.819	0.826
	GPUDMM	0.861	0.894	0.894
	BTM	0.905	0.918	0.929
	NBTM	0.718	0.747	0.758
	NBTMWE	**0.930**	**0.937**	**0.944**
Web Snippets	DMM	0.849	0.849	0.856
	LFDMM	0.794	0.788	0.789
	GPUDMM	0.860	0.865	**0.872**
	BTM	0.774	0.774	0.778
	NBTM	0.655	0.689	0.696
	NBTMWE	**0.865**	**0.871**	0.871

4.6 Efficiency

To investigate the efficiency of the proposed method, we compare the running time of all models implemented in Java. Table 6 shows the average time cost per iteration of the sampling process in each of the methods with $K = \{40, 60, 80\}$. From the results, we find that BTM and DMM are much more efficient than their word-embeddings-based extended models. It is because in each iteration, the later models need to cost additional time on biterm-topic probability (or word-topic probability in GPUDMM) updating and topic counter updating of the related words.

In addition, as for NBTMWE, increasing the number of topics also increases the time cost on sampling process while improves the classification accuracy slightly. In fact, the accuracy increases to stability while the topic coherence will decease when continue to enlarge the number of latent topics. Therefore, it needs to set an appropriate value for the number of topics to balance the metrics of topic coherence, text classification accuracy and time cost in practical application.

4.7 Parameter tuning

We now study the impact of parameter μ on the model performance. Higher value of μ means higher topic promotion degree for biterm related words during the sampling process.

Figure 9 shows the classification accuracy, topic word Coherence and WESim with K=60 and N=10. From the results, we can see the accuracy achieves the best performance when μ set to a small value while WESim achieves the best performance when μ set to a high value on both datasets. It is because when μ is low (e.g., μ =0.2), namely the topic promotion degree of biterm related words is low, the biterm related words slightly impact on the topic inference. When further enlarge μ, the average word similarity among latent topic words improves significantly while the accuracy reduces slightly. It is because when μ is high (e.g., μ=0.9), the model is apt to cluster the semantically similar words into the same topic while it considers less about whether these words co-occur in the text of the dataset.

Table 6 Average time cost (second) per iteration of different methods testing on both datasets

Dataset	Methods	K=40	K=60	K=80
Sina Weibo	DMM	0.245	0.417	0.577
	LFDMM	19.107	28.566	35.887
	GPUDMM	0.423	0.592	0.759
	BTM	0.233	0.413	0.676
	NBTM	0.264	0.428	0.622
	NBTMWE	0.309	0.538	0.813
Web Snippets	DMM	0.024	0.042	0.068
	LFDMM	7.513	10.27	12.464
	GPUDMM	0.095	0.112	0.119
	BTM	0.206	0.31	0.431
	NBTM	0.237	0.329	1.241
	NBTMWE	0.334	0.47	0.621

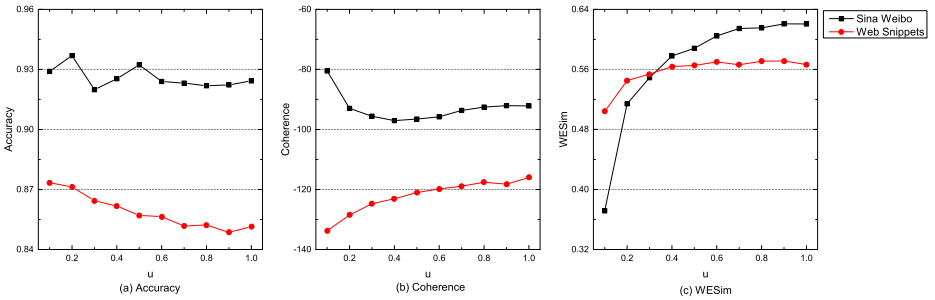


Figure 9 Effect of parameter μ on WESim and classification accuracy under K=60 and N=10

5 Conclusions

Due to the sparsity of short texts, conventional topic models usually face the challenges of incoherent and unintelligible of topic representation. In this study, we proposed a novel topic model, named NBTMWE, which combines word embeddings pre-trained from external corpus with Noise Biterm Topic Model for topic inference. This model enhances topic coherence by introducing a noise topic into the model and further promoting both word co-occurrence patterns and semantically similar words into the same meaningful topic. We conduct extensive experiments on two real-world short text datasets. The experimental results show that NBTMWE outperforms state-of-the-art methods from topic coherence and text classification accuracy. Furthermore, this model still can be improved from several views in the future. For example, similar biterms evaluated from embeddings can be promoted to the same topics, which is not considered in this work. Other improvement is to jointly train word embeddings with latent topics from biterm view. Via these strategies, we expect to further improve the topic coherence and classification accuracy, especially on the dataset that consists of similar issues.

Acknowledgements The work was supported by the National Natural Science Foundation of China (NSFC, No.61802194, No.61902190) and Natural Science Foundation in University of Jiangsu Province, China (No.17KJB520015, No.19KJB520040).

References

1. Amplayo, R.K., Lee, S., Song, M.: Incorporating product description to sentiment topic models for improved aspect-based sentiment analysis. Inf. Sci. **454-455**, 200–215 (2018)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J Mach Learn Res **3**, 993–1022 (2003)
3. Chen, Z., Liu, B.: Topic modeling using topics from many domains, lifelong learning and big data. In: International conference on machine learning, pp. 703–711 (2014)
4. Chen, Z., Liu, B.: Mining topics in documents: standing on the shoulders of big data. In: ACM conference on knowledge discovery and data mining, pp. 1116–1125 (2014)
5. Cheng, X., Yan, X., Lan, Y., Guo, J.: BTM: Topic modeling over short texts. IEEE Trans Knowl Data Eng **26**(12), 2928–2941 (2014)
6. Das, R., Zaheer, M., Dyer, C.: Gaussian LDA for topic models with word embeddings. In: International joint conference on natural language processing, pp. 795–804 (2015)
7. Fang, A., Macdonald, C., Ounis, I., Habel, P.: Using word embedding to evaluate the coherence of topics from twitter data. In: ACM SIGIR conference on research and development in information retrieval, pp. 1057–1060 (2016)

8. Gao, W., Peng, M., Wang, H., Zhang, Y., Xie, Q., Tian, G.: Incorporating word embeddings into topic modeling of short text. Knowl. Inf. Syst. **61**(2), 1123–1145 (2019)
9. Garciapablos, A., Cuadros, M., Rigau, G.: W2VLDA: Almost unsupervised system for aspect based sentiment analysis. Expert Systems with Applications: 127-137 (2018)
10. Haigh, J.: Pólya urn models. Journal of the Royal Statistical Society:, Series A (Statistics in Society) **172**(4), 942–942 (2009)
11. Hofmann, T.: Probabilistic latent semantic indexing. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)
12. Huang, J., Peng, M., Wang, H., Cao, J., Gao, W., Zhang, X.: A probabilistic method for emerging topic tracking in microblog stream. World Wide Web **20**(2), 325–350 (2017)
13. Jiang, L., Lu, H., Xu, M., Wang, C.: Biterm pseudo document topic model for short text. In: IEEE International Conference on Tools with Artificial Intelligence, pp. 865–872 (2016)
14. Li, S., Chua, T.S., Zhu, J., Miao, C.: Generative topic embedding: a continuous representation of documents. In: Annual Meeting of the Association for Computational Linguistics, pp. 666–675 (2016)
15. Li, C., Duan, Y., Wang, H., Wang, H., Zhang, Z., Sun, A., Ma, Z.: Enhancing topic modeling for short texts with auxiliary word embeddings. ACM Trans. Inf. Syst. **36**(2), 30 (2017)
16. Li, C., Wang, H., Zhang, Z., Sun, A., Ma, Z.: Topic modeling for short texts with auxiliary word embeddings. In: International ACM SIGIR conference on research & development in information retrieval, pp. 165–174 (2016)
17. Mehrotra, R., Sanner, S., Buntine, W., Xie, L.: Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In: International ACM SIGIR conference on research & development in information retrieval, pp. 889–892 (2013)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: International Conference on Learning Representations, pp. 1–12 (2013)
19. Mimno, D.M., Wallach, H.M., Talley, E.M., Leenders, M., Mccallum, A.: Optimizing semantic coherence in topic models. In: Conference on empirical methods in natural language processing, pp. 262–272 (2011)
20. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: Human language technologies: Conference of the North American chapter of the association of computational linguistics, pp. 100–108 (2010)
21. Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M.: Improving topic models with latent feature word representations. Trans. Assoc. Comput. Linguistics **3**, 299–313 (2015)
22. Peng, M., Chen, D., Xie, Q., Zhang, Y., Wang, H., Hu, G., Gao, W., Zhang, Y.: Topic-net conversation model. In: International conference on Web information systems engineering, pp. 483–496 (2018)
23. Peng, M., Gao, B., Zhu, J., Huang, J., Yuan, M., Li, F.: High quality information extraction and query-oriented summarization for automatic query-reply in social network. Exp. Syst. Appl. **44**(2016), 92–101 (2016)
24. Peng, M., Huang, J., Fu, H., Zhu, J., Zhou, L., He, Y.: High quality microblog extraction based on multiple features fusion and time-frequency transformation. In: International conference on web information systems engineering, pp. 188–201 (2013)
25. Peng, M., Xie, Q., Zhang, Y., Wang, H., Zhang, X., Huang, J., Tian, G.: Neural sparse topical coding. In: Annual meeting of the association for computational, pp. 2332–2340 (2018)
26. Peng, M., Zhu, J., Wang, H., Li, X., Zhang, Y., Zhang, X., Tian, G.: Mining event-oriented topics in microblog stream with unsupervised multi-view hierarchical embedding. ACM Trans. Knowl. Discov. Data **12**(3), 1–26 (2018)
27. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & Web with hidden topics from large-scale data collections. In: International Conference on World Wide Web, pp. 91–100 (2008)
28. Quan, X., Kit, C., Ge, Y., Pan, S.J.: Short and sparse text topic modeling via self-aggregation. In: International conference on artificial intelligence, pp. 2270–2276 (2015)
29. Shi, B., Lam, W., Jameel, S., Schockaert, S., Lai, K.P.: Jointly learning word embeddings and latent topics. In: International ACM SIGIR conference on research & development in information retrieval, pp. 375–38 (2017)
30. Sorokin, D., Gurevych, I.: Context-aware representations for knowledge base relation extraction. In: Conference on Empirical Methods in Natural Language Processing, pp. 1784–1789 (2017)
31. Sun, A.: Short text classification using very few words. In: International ACM SIGIR conference on research & development in information retrieval, pp. 1145–1146 (2012)
32. Wen, J., Tu, H., Cheng, X., Xie, R., Yin, W.: Joint modeling of users, questions and answers for answer selection in CQA. Exp. Syst. Appl. **118**(2019), 563–572 (2019)

33. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: Finding topic-sensitive influential Twitterers. In: ACM international conference on Web search and data mining, pp. 261–270 (2010)
34. Wu, Z., Lei, L., Li, G., Huang, H., Zheng, C., Chen, E.: A topic modeling based approach to novel document automatic summarization. Exp. Syst. Appl. **84** (2017), 12–23 (2017)
35. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: International conference on World Wide Web, pp. 1445–1456 (2013)
36. Yan, X., Guo, J., Lan, Y., Xu, J., Cheng, X.: A probabilistic model for bursty topic discovery in microblogs. In: AAAI conference on artificial intelligence, pp. 353–359 (2015)
37. Yang, Y., Wang, F., Zhang, J., Xu, J., Yu, P.S.: A topic model for co-occurring normal documents and short texts. World Wide Web **21**(2), 487–513 (2018)
38. Yin, J., Wang, J.: A Dirichlet multinomial mixture model-based approach for short text clustering. In: ACM SIGKDD international conference on knowledge discovery & data mining, pp. 233–242 (2014)

Affiliations

Jiajia Huang[1] · Min Peng[2] · Pengwei Li[1] · Zhiwei Hu[3] · Chao Xu[1]

Jiajia Huang
huangjj@nau.edu.cn

Min Peng
pengm@whu.edu.cn

Zhiwei Hu
zhiweihu@whu.edu.cn

Chao Xu
xuchao@nau.edu.cn

1 Nanjing Audit University, Nanjing, 211815, China

2 Wuhan University, Wuhan, 430072, China

3 Shanxi Agricultural University, Datong, 030801, China