GPU-BTM: A Topic Model for Short Text using Auxiliary Information

Yibing Guo[1], Yutao Huang[1], Ye Ding[2], Shuhan Qi[1], Xuan Wang[1], Qing Liao[1,3]✉
[1]School of Computer Science and Technology, Harbin Institute of Technology (shenzhen), Shenzhen, China
[2]School of Computer Science and Technolog, Dongguan University of Technology, Dongguan, China
[3]Peng Cheng Laboratory, Shenzhen, 518000, China
Email: 18s151529@stu.hit.edu.cn, yutaohuang7777@163.com, dingye@dgut.edu.cn,
{shuhanqi, wangxuan}@cs.hitsz.edu.cn, ✉liaoqing@hit.edu.cn

***Abstract*—Recently, short texts become very popular in social life. To understand short texts, researchers develop topic models to extract topic information. However, conventional topic models mainly focus on long documents which cannot deal with the sparsity problem of short text. In this paper, we propose a novel topic model for short text called GPU-BTM, which incorporates Generalized Pólya Urn technique into Biterm Topic Model. GPU-BTM utilizes the similarity information and the co-occurrence pattern of words simultaneously to handle the sparsity problem. Specifically, the GPU module considers the similarity information among words, so that GPU-BTM generates more coherent topics. On the other hand, BTM module tries to capture the co-occurrence pattern of words so that the enriched contexts relieve the data sparsity problem. In the experiment part, the results demonstrate that GPU-BTM model outperforms four latest comparison models on two real world short text datasets.**

***Index Terms*—short text, topic model, auxiliary information**

I. Introduction

Since short texts are quite popular in various forms like short messages, microblogs and news titles, it is very important to extract their topic information. Actually, researchers design topic models to understand and summarize short texts. Theoretically, topic models aim to extract topics from texts and do topic assignment on texts in an unsupervised manner. Because of the ability to extract topic information from text data, topic models attract more and more researchers to do related investigation.

There already are lots of helpful topic models for several decades. However, conventional topic models like Latent Dirichlet Allocation (LDA) [1] have quite discrepant performances on short texts and long documents. A long document usually contains thousands of words while a short text only contains dozens of words or even less. Short texts suffer from severe data sparsity problem due to its lacking of enough context [2]. That is why conventional topic models fail to extract coherent topics from short texts.

To improve the performance of topic models on short texts, many strategies are proposed to focus on short texts. Pseudo-document strategy aims to assemble hundreds of short texts into a long document according to some extra rules like same authors [3]. A strong assumption strategy [4] leads to better topic extraction by adding some extra restrictions on short texts. A typical restriction is to require that there is only one topic in each short text. However, the two strategies are considered less helpful because they perform badly on experiments.

Another strategy concentrates on word pairs instead of a single word. Biterm Topic Model (BTM) [5] is the model which is capable to handle short texts by modeling the co-occurrence pattern of words on the whole corpus. In BTM, a word pair staying in the same sliding window is called biterm which captures the word co-occurrence pattern directly. Biterm is considered as the very design which endows BTM with the capability to relieve the sparsity problem. Even though BTM does perform better than conventional topic models on short texts, it does not generate topics with high coherence since it is restricted to the target corpus.

Besides, one of the most useful strategies is auxiliary information strategy. Such a strategy introduces extra auxiliary information to enrich the context of target short texts [6]. It is intuitive. When a person tries to understand a short text with few contexts, he can understand it by recalling some background information from his previous reading experience. According to the reading experience, the words with similar semantic relatedness or high co-occurrence pattern are more possible to share the same topics. For example, when a person finds ”apple” and ”pear” in a context, he will know that this text talks about fruit with probability. This is because the two words usually stay in same contexts and both refer to the topic ”fruit”. Inspired by such a phenomenon and considering that BTM lacks of enough information to achieve good topic modeling, we find a proper way to introduce auxiliary information into BTM to enrich the contexts of target corpus. Actually, we introduce Generalized Pólya Urn (GPU) [7], a wildly-used probability model. By incorporating GPU into BTM, we emphasize that GPU-BTM is quite possible to assign the same topics to the words with high similarity. More specifically, every time a biterm is assigned with a topic, the similar words union of the two inner words are assigned with the same topic assignment via GPU. In this paper, we introduce a novel topic model for short texts called Generalized Pólya Urn Biterm Topic Model (GPU-BTM).

In our model GPU-BTM, auxiliary information is auxiliary corpus with millions of texts. We utilize a word embedding model like GloVe [8] to learn word vectors from auxiliary

978-1-7281-9558-2/20/$31.00 ©2020 IEEE
DOI 10.1109/DSC50466.2020.00037

corpus. Then the similarity of any two words will be obtained by calculating the cosine similarity of their vectors representation. Instead of modeling words like other models do, GPU-BTM directly models the co-occurrence pattern of words on the whole target corpus. The similarity information will be introduced into the topic inference process to guide GPU-BTM to promote the probability of similar words assigned to same topics. GPU-BTM is able to generate highly coherent topics and do more reasonable topic assignment due to its considering on the co-occurrence pattern of words and auxiliary information simultaneously.

The main contributions of this paper are concluded as below.

- We introduce a novel topic model for short text called GPU-BTM which is able to handle the data sparsity problem effectively. Our model produces highly coherent topics by introducing auxiliary information into BTM via GPU method. In GPU-BTM, GPU scheme properly improve the important information of biterms by considering the similar words unions of inner words, which makes GPU-BTM performs quite well.
- We do experiments on real world datasets to evaluate GPU-BTM and other 4 comparison topic models in terms of topic coherence and short text classification. According to the experimental results, GPU-BTM does outperform other models.

II. Related Work

A. Topic Model for Long Documents

Previous topic models mainly focus on extracting topic distributions from long documents. Famous conventional topic models include Latent Semantic Analysis (LSA) [9], Probabilistic Latent Semantic Indexing (PLSI) [10], LDA, and Hierarchical Dirichlet Process (HDP) [11]. LSA converts a document into a document-word matrix and then decomposes it into low-dimensional latent space representation which can be treated as topics via Singular Value Decomposition (SVD) [12]. However, LSA suffers from the issue of polyseme and SVD is quite time-consuming. PLSI, a generative model which can be solved by EM algorithm [13], comes up with the concepts of topic-word distribution and document-topic distribution for the first time. It is a milestone in the development process of topic model for binging in the two concepts. Besides, by introducing Dirichlet prior and Bayesian knowledge into PLSI, LDA becomes one of the most popular topic models due to its good performance. The previous models have to choose an important hyperparameter, the number of topics, manually, which makes the models less intelligent. In order to solve this problem, researchers introduced a non-parametric Bayesian method called HDP to determine the number of topics automatically based on Dirichlet process. However such models are not able to generate coherent topics or do topic assignment effectively on short texts due to the data sparsity problem of short texts.

B. Topic Model for Short Texts

Since short texts differ from long documents on the average length, researchers design several special categories of topic models to handle the sparsity problem of short texts. The first category is to apply strong assumptions on short texts to reduce the diversity of topics. The second category is to model different patterns in the corpus instead of a single word. The third category is to introduce auxiliary information to enrich the contexts of short texts.

For the first category, models are designed to apply extra strong assumption on short texts to reduce the diversity of topics. Mixture of unigram model (MU) [14] assumes that there is only one topic in a text. It means that all the words in a short text follow the same topic-word distribution. By restricting LDA with the condition that there is only one chance for prior topic drawing during text generation, Dirichlet Multinomial Mixture Model (DMM) [15] has better performance than LDA on short texts. However, Such models cannot model the real data properly due to the information loss caused by hard conditions.

For the second category, BTM is quite typical. Different from previous ones, BTM handles the data sparsity problem by modeling the co-occurrence pattern of word instead of a single word. Besides, it abandons the concept of text and tries to do topic inference on the whole target corpus. Another novel model called Attentional Segmentation Topic Model (ASTM) [16] achieves outstanding performance by simulating the reading habits of human beings. The reading habits include the background information and text's segmentation into smaller phrases. Topic models in this category attempt to understand short texts from a novel perspective. However, without enough information to help interpret the texts, such models hardly perform satisfactorily.

The third category is proved to be one of the most useful categories. Since short texts lack of enough contexts for topic modeling, introducing auxiliary information will be helpful. Now that short texts are short in length, some topic models try to assemble short texts into long documents according to some auxiliary information like author. After the assembling work, conventional topic models are utilized to do topic extraction and assignment. Typical examples are Pseudo-Document-Based Topic Model (PTM) [17] and Self-Aggregate-Based Topic Model (SATM) [18]. However, both PTM and SATM heavily depends on their assembling settings. To avoid the problem, other models choose to use auxiliary information in different ways. For example, Latent Feature Topic Model (LFTM) generates words according to a combination of a topic-word distribution component and a word latent representation component [19]. GPU-DMM [20] combines GPU and DMM, which only models important words for topic extraction. The important words are estimated by the introduced auxiliary information. We were inspired by this model to come up with GPU-BTM which goes beyond GPU-DMM by concentrating on the co-occurrence pattern of words.

C. Incorporating Word Embedding into Topic Model

Word embedding is a good method to introduce auxiliary information. It is utilized to extract the information of word representation from auxiliary corpora. For example, the model mentioned above, ASTM, enriches the background information of short text with word embedding. Besides, Latent Concept Topic Model (LCTM) [21] models a topic with a Gaussian distribution on word embedding space. In LCTM, mathematic methods are more reasonable to be introduced due to the conversion from plain words to word vectors. Skip-gram Topical word Embedding (STE) [22] uses skip-gram architecture of word2vec [23] to introduce auxiliary information via word embedding. Semantic Assisted Non-negative Matrix Factorization (SeaNMF) [27] integrates semantic information into NMF method using word embedding. In this way, SeaNMF obtains a more helpful representation matrix to achieve better topic modeling work. Embedding-based Topic Model (ETM) [28] uses word embedding with semantic information to aggregate short texts into pseudo-texts. Then ETM introduces Markov random field canonical model to give related words a better chance of being applied to the same topic.

Our model GPU-BTM also applies word embedding to import auxiliary information. More specifically, we learn word vector representation from auxiliary corpus via GloVe. By calculating the cosine similarity of these word vectors, GPU-BTM obtains similarity information among words. Similarity information is the key to generate more coherent topics.

III. GPU-BTM

GPU-BTM incorporates GPU into BTM to strengthen the whole model's ability to extract more coherent topics. Modeling on the co-occurrence pattern of words helps relieve the data sparsity problem. Utilizing GPU to improve the probability of similar words assigned to same topics during topic inference process make the topic more coherent. By combining the co-occurrence pattern of words and auxiliary information, GPU-BTM produces highly coherent topics and does topic assignment on each short text effectively. In the following part, GPU-BTM will be described in detail.

A. Biterm Topic Model

Unlike conventional topic models for long document, BTM is able to relieve the sparsity problem of short texts by modeling the co-occurrence pattern of words. Actually, this pattern is modeled by a word pair in same sliding window which is called biterm. Besides, BTM abandons the concept of a single text and models the co-occurrence pattern on the whole corpus.

More specifically, suppose that there is a corpus containing N_B biterms and W different words. For hyperparameters, the number of topics is set as K and another two Dirichlet distributions controlling hyperparameters are set as α and β respectively. The generation of a corpus can be described as the process below.

- Draw topic distribution $\theta \sim Dirichlet\,(\alpha)$.
- For the k-th topic, $k \in [1, K]$:
  Draw topic-word distribution $\Phi_k \sim Dirichlet\,(\beta)$.
- Draw the two words in biterm b_i:
  Draw the topic of b_i, $z \sim Multinomial\,(\theta)$.
  Draw the word pair $w_{i,1}, w_{i,2} \sim Multinomial\,(\theta_z)$.

According to the generation rule, the posterior of topic-word distributions and document-topic distributions, Φ and θ, can be calculated by using Gibbs Sampling algorithm. After that, the topics group and topic distribution is easy to be inferred.

B. Generalized Pólya Urn Model

Generalized Pólya Urn model is a wildly-used probability model in many fields such as loimology investigation. Suppose that there is an urn containing dozens of balls. Every ball is painted with a certain kind of color. there are several different colors totally. The process is that every time a ball is selected from the urn, a certain number of balls with similar color are put back along with the drawn ball and another same color ball. In this way, the drawn ball's color and its similar color get promoted during the drawing process.

GPU can be incorporated into the Gibbs Sampling algorithm to promote the probability that similar words being assigned to the same topic. Actually, in a topic model task, the corpus is considered as the urn while balls can be considered as words and colors can be considered as topics. Therefore, the balls with similar colors are just like the words with high similarity. During the process of topic inference, every time a word is assigned to some topic, its similar words will also be assigned with the very topic for certain times. Then words with high similarity are more probable to share same topics. Specifically, GPU-BTM uses GPU scheme to help improve the probability that words with similar semantic meaning share same topics. Besides, unlike other topic model with GPU scheme, our model based on biterm is able to consider the topic information of inner words simultaneously. In other words, GPU-BTM achieves probability improvement more accurately because it obtains similar words by getting the similar words union of both two inner words of a biterm. Such a strategy does help GPU-BTM generate more coherent topics.

In the next section, we will describe the details of GPU-BTM workflow.

C. GPU-BTM

GPU-BTM improves the original BTM by utilizing GPU to introduce auxiliary corpus into topic inference process. More specifically, word can be converted into word vector representation via word embedding. Then the similarity of words can be obtained by calculating the cosine similarity of their word embedding vectors. By setting a threshold σ on cosine similarity, every word gets its similar words. For a biterm, its similar words are considered as the union of two inner words' similar words. During the topic inference process, when a biterm is assigned to some topic, its similar words union will be assigned to the very topic for μ times. Consequently, similar words are more possible to be assigned

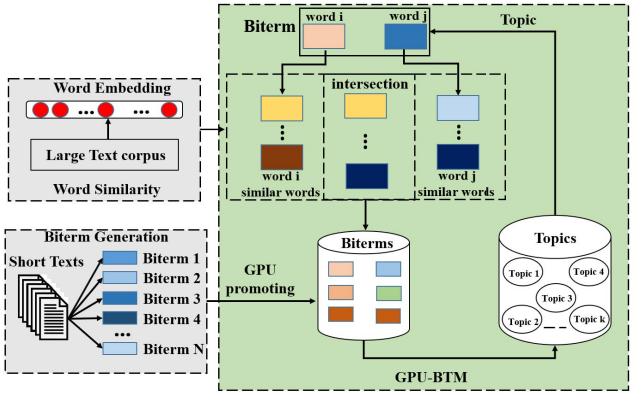


Fig. 1: GPU-BTM. Similar words are extracted from auxiliary corpus via word embedding. During Gibbs Sampling, when a biterm b_i is assigned to some topic, the words in it w_1, w_2 are assigned with the same topic for once. Besides, their "similar words" are assigned with the topic for μ times. After a certain number of iterations, the topics and topic distribution on the whole corpus will be derived.

to the same topic. The overall structure of GPU-BTM is shown in Fig.1.

Biterm Extraction. A biterm is a word pair in the same sliding window. For window "cat is cute", biterm extraction is done as below.

$$"cat \quad is \quad cute" \rightarrow \{cat, is\}, \{cat, cute\}, \{is, cute\}$$

Different window size captures the co-occurrence pattern in different degrees. In practice, a more reasonable choice is that setting the window size as a short text itself. Such a choice not only generates reasonable biterms but also makes biterm extraction more efficient. Under the condition that biterms' words come from the same short text, biterm extraction becomes the combination of any two random words in the same short text.

Similarity Calculation. Now that word embedding is obtained, GPU-BTM utilizes cosine similarity metric to calculate the similarity of two different words. For two words represented by two 100-dimensional vectors w_1 and w_2, their similarity can be calculated by (1):

$$similarity\left(w_1, w_2\right) = \frac{w_1^T \cdot w_2}{\|w_1\|\|w_2\|} \tag{1}$$

Then only words with high similarity should be introduced into topic inference process. Actually, by predefining a similarity threshold σ, words with similarity higher than the threshold are considered as similar words. In this way, a similarity matrix E of words can be obtained via (2).

$$E = \begin{cases} 1 & similarity(w_1, w_2) \geq \sigma \\ 0 & similarity(w_1, w_2) < \sigma \end{cases} \tag{2}$$

where w_1 and w_2 are any two words in the corpus.μ is a hyperparameter, the word boost amount.

GPU-BTM Workflow. First we extract the biterm set with the sliding window are set as a whole text. At the same time, collecting the similar words by calculating cosine similarity of words' word embedding representation can be done with auxiliary corpus and GloVe model. Next, Gibbs Sampling with GPU model is introduced to do model inference. What Gibbs Sampling actually does is to infer the topic assignment of every biterm and word according to the topic assignment of other biterms. The posterior information, the topic groups Φ and the topic distribution for the whole corpus θ, can be calculated according to the topic assignment of the biterms.

The probability of biterm b_i assigned to topic z can be calculated by (3) from [11].

$$\begin{aligned} &p\left(z_i = k | z_{-i}, B, \alpha, \beta\right) \propto \left(n_{-i,k} + \alpha\right) \\ &\frac{\left(n_{-i,w_{i,1}|k} + \beta\right)\left(n_{-i,w_{i,2}|k} + \beta\right)}{\left(n_{-i,\cdot|k} + W\beta + 1\right)\left(n_{-i,\cdot|k} + W\beta\right)} \end{aligned} \tag{3}$$

where z_{-i} is the topic assignment of all the biterms except biterm b_i, $n_{-i,k}$ is the number of biterms assigned to topic k except b_i, $n_{-i,w|k}$ is the times that word w assigned to topic k except for b_i, $n_{-i,\cdot|k} = \sum_{w=1}^{W} n_{-i,w|k}$.

Draw a topic for the biterm b_i according to the probability calculated above. If the topic z is assigned to b_i, $w_{i,1}$ and $w_{i,2}$ in b_i are also assigned with topic z. Then, according to GPU method, the similar words of $w_{i,1}$ and $w_{i,2}$ are assigned with topic z for μ times.

After obtaining the counting information for topic assignment of words, we can calculate the posterior information by (4) and (5).

$$\phi_{k,w} = \frac{n_{w|k} + \beta}{n_{\cdot|k} + W\beta} \tag{4}$$

$$\theta_k = \frac{n_k + \alpha}{N_B + K\alpha} \tag{5}$$

The Gibbs Sampling algorithm of GPU-BTM is shown below. The algorithm's input includes the biterm set B, the word boost amount μ, prior information α and β. The output of the algorithm are posterior information, the topic-word distributions Φ and the document-topic distributions θ.

Now that the posterior information Φ and θ are calculated, the topic distribution of each short text d with N_d biterms $\left\{{b_i}^d\right\}^{N_d}$, $P\left(z|d\right)$, can be derived.

$$\begin{aligned} P\left(z|d\right) &= \sum_{i=1}^{N_d} P\left(z, {b_i}^d | d\right) \\ &= \sum_{i=1}^{N_d} P\left(z|{b_i}^d, d\right) P\left({b_i}^d | d\right) \end{aligned} \tag{6}$$

Where

$$P\left(z|{b_i}^d, d\right) = P\left(z|{b_i}^d\right) \tag{7}$$

$$P\left({b_i}^d | d\right) = \frac{n\left({b_i}^d\right)}{\sum_{i=1}^{N_d} n\left({b_i}^d\right)} \tag{8}$$

$$P\left(z = k|{b_i}^d\right) = \frac{\theta_k {\phi^d}_{k,w_{i,1}} {\phi^d}_{k,w_{i,2}}}{\sum_{k=1}^{K} \theta_k {\phi^d}_{k,w_{i,1}} {\phi^d}_{k,w_{i,2}}} \quad (9)$$

Algorithm 1 Gibbs Sampling Algorithm for GPU-BTM

Input: the number of topics K, hyperparameters α, β, word boost amount μ, biterm set B, word similarity matrix E
Output: Φ and θ

```
initialize topic assignments randomly for all the biterms
for each iter ∈ [1, N_iter] do
  for each b ∈ B do
    get w_1, w_2, z
    get w_1, w_2 similar word intersection I from E
    for each w' ∈ I do
      update n_{w'|z} = n_{w'|z} − μ
    end for
    update n_{w_1|z} − = 1, n_{w_2|z} − = 1, n_z − = 1
    draw z_new from (1), and z = z_new
    update n_{w_1|z} + = 1, n_{w_2|z} + = 1, n_z + = 1
    for each w' ∈ I do
      update n_{w'|z} = n_{w'|z} + μ
    end for
  end for
end for
compute Φ in (4) and θ in (5)
```

IV. EXPERIMENT

In this section, we do experiments on real world datasets to compare our model with several comparison models in terms of topic coherence and text classification. The experimental results fully demonstrate the superiority of GPU-BTM.

Since there are two parts, topic-word distributions and document-topic distributions, in the outcomes of topic models, two different experiments are designed to evaluate the models' correspondingly. For topic coherence evaluation experiment, we compare the ability of topic models to generate coherent topics via metric PMI-Score [24]. For text classification experiment, we try to figure out the accuracy of each topic model's topic assignment by doing classification on the vectorized texts.

A. Datasets

Web Snippet: Web Snippet is a dataset extracted from web search engines. It contains 12340 web search snippets, where an average length of each snippet is 20 words. All the snippets in this dataset are classified into 8 categories by their original labels.

Sentiment140: Sentiment140 contains 1,600,000 tweets which are extracted from Twitter using Twitter's official API. In our work, Sentiment140 is introduced to do PMI-Score evaluation. Therefore, we only keep the text part of tweets.

To make the experiments more convincing, we determined to preprocess the datasets with the following operations: (1) convert all letters to lowercase, (2) delete words that appear less than 3 times in the dataset, (3) remove words of length less than 2, (4) delete the stop words in the text.

After preprocessing, the detail statistics information of the two datasets is shown in TABLE 1.

TABLE 1: brief summary of datasets

DataSet	Categories	Docs	Average Words	Vocabulary Size
snippets	8	11382	12.0948	6860
sentiment140	2	1599939	9.2511	115031

B. Baselines

- **BTM** is one of the most useful topic models to handle topic modeling of short text. It tries to do topic modeling based on the co-occurrence pattern of words from corpus. BTM assumes that the two words in a biterm share same topics. BTM is proved to have good and stable performances on various short text datasets.
- **DMM** is a classic topic model for short texts based on Dirichlet Multinomial Mixture model. In DMM, each short text is considered to own only one topic. After the topic is confirmed, all the words in the document should follow the topic-word distribution of the very topic.
- **GPU-DMM** is designed to extend DMM by incorporating auxiliary information via GPU scheme in 2016. Just like GPU-BTM, it introduces external corpus via word embedding to enrich the contexts of short texts. Different from our model, GPU-DMM incorporates the auxiliary information directly into the words' topic assignment. It is one of the state-of-the-art topic models for short texts.
- **R-BTM** which was published in 2019 [25] is designed base on BTM. R-BTM uses similar words of biterms to generate more external biterms by incorporating auxiliary information.

C. Experiment Configuration

To make the comparison equivalent, we set the Dirichlet hyperparameters for all the models as α = 50/K, β = 0.01. For the Gibbs sampling process of each model in the experiment, we set the iteration time as 1,000 times. Besides, other parameters for each model are set according to their original paper.

Word Embedding. Our model use GloVe to do word embedding. There already are available 100-dimensional word embeddings trained on the Wikipedia 2014 and Gigaword 5 provided by Stanford. The similarity threshold is set it as 0.6. Besides, only the top 20 words with the highest similarity for the target word are reserved if there are too many words higher than the threshold. Such a setting leads to an efficient and validate experiment for GPU-BTM. GPU word boost amount is set as 0.1 empirically. GPU-DMM, according to the settings of the original paper, a 300-dimensional word embedding pretrained on Google News corpus provided by Google is used. For R-BTM, a 100-dimensional word embedding of word2vec of CBOW architecture is used. Similarity threshold is set it as 0.6. Besides, only the top 20 words with the highest similarity

for the target word are reserved if there are too many words beyond the threshold just like our model.

To make the experiment more convincing, all the experiment results of each topic model are taken from an average of 10 rounds.

D. Evaluation of Topic Coherence

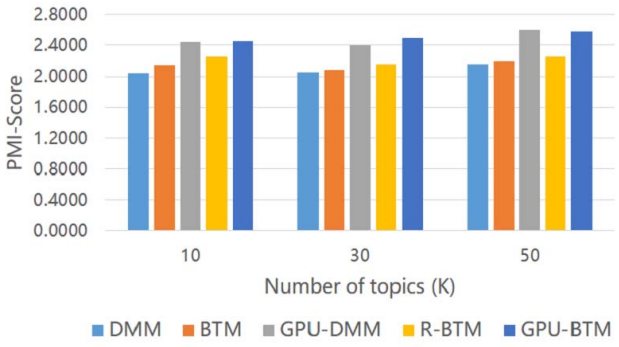
Perplexity is a classic NLP metric to evaluate the short texts and their topics. However, perplexity is severely influenced by the capacity of datasets and preprocessing. Besides, perplexity does not concentrate on the topic coherence. In our experiments, the PMI-Score is used to evaluate topic coherence. PMI-Score is a metric which can evaluate topic coherence in a way quite consistent with human being. It evaluates whether the word distribution in a target topic is consistent in an external corpus, which is also known as topic coherence. A higher PMI-Score indicates more coherent topics. PMI-Score is the most popular metric in topic coherence evaluation experiments. By showing PMI-Score, we can tell the models with a better ability to generate coherent topics. The formula of PMI-Score is showed as follows:

$$PMI(k) = \frac{2}{T(T-1)} \sum_{1 \le i \le j \le T} \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \qquad (10)$$

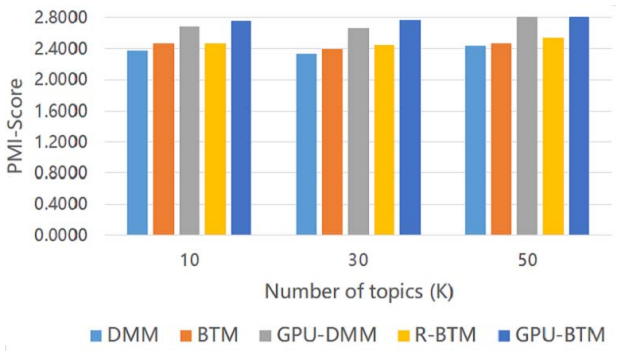
In the formula, k refers to the k-th topic, T refers to the number of words with the greatest probability in the topic-distribution of topic k. $p(w_i)$ indicates the number of times that word w_i appears in the document, and $p(w_i, w_j)$ is the number of times that words w_i and w_j appear together. In the experiments, we set total topic numbers K as $\{10,30,50\}$ and T as $\{5,10,15\}$ respectively for all models.

The calculation of PMI-Score involves two different datasets. We first apply the five topics model on Web Snippet dataset to extract topics from a target dataset. Then we count the number of times that words of top T in each topic showing in Sentiment140 dataset.

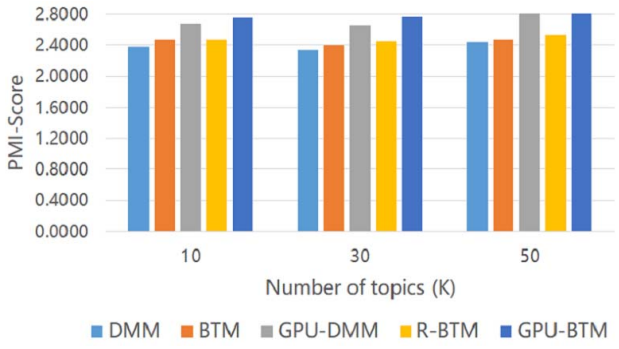
As shown in Fig.2, our model achieves the best results in most cases and has a significant improvement over the other models. GPU-DMM model stays on the second place. In some settings, GPU-DMM achieves similar performances with GPU-BTM. Such a result is reasonable. Both GPU-BTM and GPU-DMM introduces auxiliary information via GPU scheme. Therefore, the two topic models gain the ability to enrich the contexts of target corpus. However, GPU-BTM models the co-occurrence pattern of words on the whole corpus instead of a single word in a text. GPU-BTM considers the similar words of both words in a biterm while GPU-DMM merely considers one single word's similar words. Therefore, GPU-BTM is able to use GPU scheme more accurately and effectively. Such a difference makes the different performances of GPU-BTM and GPU-DMM in terms of topic coherence. R-BTM achieves worse performance than the two models with GPU scheme. Although R-BTM introducing auxiliary information as well, it simply generates new biterms manually using similar words of each short text. The way to enrich the contextual information of R-BTM introduces too much new noise information, since the new biterms are considered equally as original biterms which covers the original topic information. It is unacceptable. By comparing with R-BTM, we can see that GPU model is an excellent choice to incorporate auxiliary information. Among other baseline models, even though BTM and DMM achieve bad topic modeling, BTM outperforms DMM obviously. Again, it proves that doing topic modeling on biterms leads to more coherent topics than on a single word.



(a) T = 5



(b) T = 10



(c) T = 15

Fig. 2: the result of topic coherence evaluation experiment on Web Snippet

E. Evaluation of Short Text Classification

In topic model tasks, the texts can be represented with their document-topic distribution vectors. By classifying the texts

according to their vector representation, we show the quality of the document-topic distributions of different topic models. A higher classification accuracy means that the topic model assigns topics to texts more accurately. In the text classification experiment, for fairness, the classifier used is fixed as Support Vector Machine (SVM) [26] for all the topic models. It is a famous open source toolkit available in scikit-learn.

We use Web Snippet dataset as the target dataset. 70% of the dataset is used as a training dataset to train the classifier. Another 30% of the dataset is set as the test dataset to evaluate the classification accuracy. The partition of Web Snippet is done randomly. After setting some parameters of classifier as default values, we use a 10-fold cross-validation method to train the model. Then the well-trained SVM classifier is utilized to do classification on the test dataset.

The results of classification experiment are shown in TABLE 2. GPU-BTM has the highest classification accuracy and it owns an obvious edge over the other comparison models. GPU-DMM model achieves the second-best result again, which demonstrates that GPU strategy is a good method to improve the quality of topic assignment. Unlike GPU-DMM, our model GPU-BTM does topic assignment considering the topic information of both words in a biterm. Since we calculate the document-topic distribution according to the topic assignment of words, our model which generates more coherent topics are supposed to generate more accurate document-word distribution. Just as we discussed, GPU-BTM goes beyond GPU-DMM due to its more accurate and effective way to use GPU scheme. R-BTM performs worse than GPU-DMM because it just introduces some noise. The external biterms cover the topic information of original biterms in the target corpus. Therefore, the performance of R-BTM is obviously worse than the previous two models with GPU scheme. Baselines without auxiliary information perform worst in this comparison. Even though both DMM and BTM don't perform well, DMM is worse than BTM in the text classification experiment. BTM does more accurate classification due to its richer contexts and modeling on the co-occurrence pattern of words.

Consequently, our model GPU-BTM gets the best performance on both two experiments. The experimental results demonstrates GPU-BTM's superiority clearly.

TABLE 2: the result of text classification evaluation experiment

DataSet	Topic model	K=10	K=30	K=50
Snippet	DMM	0.7522	0.8712	0.8892
	BTM	0.7564	0.8542	0.8689
	GPU-DMM	*0.7856*	*0.8861*	*0.8959*
	R-BTM	0.7631	0.8758	0.8735
	GPU-BTM	**0.7901**	**0.8923**	**0.9058**

V. CONCLUSION

We propose a novel topic model for short text called GPU-BTM in this article. By incorporating GPU into BTM, GPU-BTM successfully relieves the sparsity problem of short texts. GPU-BTM introduces auxiliary corpus and models the co-occurrence pattern of words simultaneously, which helps GPU-BTM to generate highly coherent topics and assign topics quite accurately. More specifically, BTM module in GPU-BTM aims to model the co-occurrence pattern of words on the whole target corpus. Then our model introduces auxiliary corpus into topic inference process of BTM via GPU method, which promotes the probability that similar words are assigned to the same topic. In this way, GPU-BTM gets abundant information to generate highly coherent topics and do text classification accurately. We do experiments to evaluate GPU-BTM and several other topic models in terms of topic coherence and text classification. The results of experiments fully demonstrate that GPU-BTM outperforms the other models.

ACKNOWLEDGMENT

This work is partly supported by National Key Research and Development Program of China (2017YFB0202201) and National Natural Science Foundation of China (No.U1711261 and No.61702134).

REFERENCES

[1] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent Dirichlet Allocation." Journal of Machine Learning Research 3: 993–1022.

[2] Hua, Wen, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2015. "Short Text Understanding through Lexical-Semantic Analysis." In 2015 IEEE 31st International Conference on Data Engineering, 495–506.

[3] Jiang, Lan, Hengyang Lu, Ming Xu, and Chongjun Wang. 2016. "Biterm Pseudo Document Topic Model for Short Text." In 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), 865–72.

[4] Yu, Jia, and Lirong Qiu. 2019. "ULW-DMM: An Effective Topic Modeling Method for Microblog Short Text." IEEE Access 7: 884–93.

[5] Yan, Xiaohui, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. "A Biterm Topic Model for Short Texts." In Proceedings of the 22nd International Conference on World Wide Web, 1445–56.

[6] Jin, Ou, Nathan N. Liu, Kai Zhao, Yong Yu, and Qiang Yang. 2011. "Transferring Topical Knowledge from Auxiliary Long Texts for Short Text Clustering." In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 775–84.

[7] Benaïm, Michel, Itai Benjamini, Jun Chen, and Yuri Lima. 2015. "A Generalized Pólya's Urn with Graph Based Interactions." Random Structures and Algorithms 46 (4): 614–34.

[8] Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "Glove: Global Vectors for Word Representation." In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–43.

[9] Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. "Indexing by Latent Semantic Analysis." Journal of the Association for Information Science and Technology 41 (6): 391–407.

[10] Hofmann, Thomas. 1999. "Probabilistic Latent Semantic Indexing." In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 51:50–57.

[11] Griffiths, Thomas L., Michael I. Jordan, Joshua B. Tenenbaum, and David M. Blei. 2003. "Hierarchical Topic Models and the Nested Chinese Restaurant Process." In Advances in Neural Information Processing Systems 16, 17–24.

[12] Candès, Emmanuel J., Xiaodong Li, Yi Ma, and John Wright. 2011. "Robust Principal Component Analysis." Journal of the ACM 58 (3): 11.

[13] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." Journal of the Royal Statistical Society Series b-Methodological 39 (1): 1–22.

[14] Wang, Xuerui, and Andrew McCallum. 2005. “A Note on Topical N-Grams.”
[15] Neal, Radford M. 2000. “Markov Chain Sampling Methods for Dirichlet Process Mixture Models.” Journal of Computational and Graphical Statistics 9 (2): 249–65.
[16] Wang, Jiamiao, Ling Chen, Lu Qin, and Xindong Wu. 2018. “ASTM: An Attentional Segmentation Based Topic Model for Short Texts.” In 2018 IEEE International Conference on Data Mining (ICDM), 577–86.
[17] Zuo, Yuan, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. 2016. “Topic Modeling of Short Texts: A Pseudo-Document View.” In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2105–14.
[18] Quan, Xiaojun, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. 2015. “Short and Sparse Text Topic Modeling via Self-Aggregation.” In IJCAI’15 Proceedings of the 24th International Conference on Artificial Intelligence, 2270–76.
[19] Nguyen, Dat Quoc, Richard Billingsley, Lan Du, and Mark Johnson. 2015. “Improving Topic Models with Latent Feature Word Representations.” Transactions of the Association for Computational Linguistics 3: 299–313.
[20] Li, Chenliang, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. “Topic Modeling for Short Texts with Auxiliary Word Embeddings.” In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, 165–74.
[21] Hu, Weihua, and Jun ’ichi Tsujii. 2016. “A Latent Concept Topic Model for Robust Topic Inference Using Word Embeddings.” In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 380–86.
[22] Shi, Bei, Wai Lam, Shoaib Jameel, Steven Schockaert, and Kwun Ping Lai. 2017. “Jointly Learning Word Embeddings and Latent Topics.” In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 375–84.
[23] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. “Distributed Representations of Words and Phrases and Their Compositionality.” In Advances in Neural Information Processing Systems 26, 3111–19.
[24] Newman, David, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. “Automatic Evaluation of Topic Coherence.” In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 100–108.
[25] Li, Ximing, Ang Zhang, Changchun Li, Lantian Guo, Wenting Wang, and Jihong Ouyang. 2019. “Relational Biterm Topic Model: Short-Text Topic Modeling Using Word Embeddings.” The Computer Journal 62 (3): 359–72.
[26] Chang, Chih-Chung, and Chih-Jen Lin. 2011. “LIBSVM: A Library for Support Vector Machines.” ACM Transactions on Intelligent Systems and Technology 2 (3): 27.
[27] Shi, Tian, Kyeongpil Kang, Jaegul Choo, and Chandan K. Reddy. 2018. “Short-Text Topic Modeling via Non-Negative Matrix Factorization Enriched with Local Word-Context Correlations.” In WWW 2018: International World Wide Web Conferences, 1105–14.
[28] Qiang, Jipeng, Ping Chen, Tong Wang, and Xindong Wu. 2017. “Topic Modeling over Short Texts by Incorporating Word Embeddings.” In Pacific-Asia Conference on Knowledge Discovery and Data Mining, 363–74.