Short-Text Topic Modeling via Non-negative Matrix Factorization Enriched with Local Word-Context Correlations

Tian Shi
Virginia Tech
tshi@vt.edu

Kyeongpil Kang
Korea University
rudvlf0413@korea.ac.kr

Jaegul Choo*
Korea University
jchoo@korea.ac.kr

Chandan K. Reddy
Virginia Tech
reddy@cs.vt.edu

ABSTRACT

Being a prevalent form of social communications on the Internet, billions of short texts are generated everyday. Discovering knowledge from them has gained a lot of interest from both industry and academia. The short texts have a limited contextual information, and they are sparse, noisy and ambiguous, and hence, automatically learning topics from them remains an important challenge. To tackle this problem, in this paper, we propose a semantics-assisted non-negative matrix factorization (SeaNMF) model to discover topics for the short texts. It effectively incorporates the word-context semantic correlations into the model, where the semantic relationships between the words and their contexts are learned from the skip-gram view of the corpus. The SeaNMF model is solved using a block coordinate descent algorithm. We also develop a sparse variant of the SeaNMF model which can achieve a better model interpretability. Extensive quantitative evaluations on various real-world short text datasets demonstrate the superior performance of the proposed models over several other state-of-the-art methods in terms of topic coherence and classification accuracy. The qualitative semantic analysis demonstrates the interpretability of our models by discovering meaningful and consistent topics. With a simple formulation and the superior performance, SeaNMF can be an effective standard topic model for short texts.

CCS CONCEPTS

• **Information systems** → **Document topic models**; *Document representation*; • **Computing methodologies** → **Topic modeling**; **Non-negative matrix factorization**;

KEYWORDS

Topic modeling, short texts, non-negative matrix factorization, word embedding.

ACM Reference Format:
Tian Shi, 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: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/https://doi.org/10.1145/3178876.3186009

*Jaegul Choo is the corresponding author.

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8.
https://doi.org/https://doi.org/10.1145/3178876.3186009

1 INTRODUCTION

Everyday, large amounts of short texts are generated, such as tweets, search queries, questions, image tags, ad keywords, headlines, and others. They have played an important role in our daily lives. Discovering knowledge from them becomes an interesting yet challenging research task which has gained a lot of attention [8, 23, 24, 26, 28]. Since short texts have only a few words, they can be arbitrary, noisy and ambiguous. All these factors make it difficult to effectively represent short texts and discover knowledge from them.

Traditionally, topic modeling has been widely used to automatically uncover the hidden thematic information from the documents with rich content [1, 5, 7]. Generally speaking, there are two groups of topic models, i.e., generative probabilistic models, such as latent Dirichlet allocation (LDA) [1], and non-negative matrix factorization (NMF) [14]. The NMF-based models learn topics by directly decomposing the term-document matrix, which is a bag-of-word matrix representation of a text corpus, into two low-rank factor matrices. The NMF based models have shown outstanding performance in dimension reduction and clustering [3, 11, 13] for the high-dimensional data.

Although the conventional topic models have achieved great success for regular-sized documents, they do not work well on short text collections. Since a short text only contains a few meaningful keywords, the word co-occurrence information is difficult to be captured [8, 28]. In the last few years, many efforts have been dedicated to tackle this challenge. A popular strategy is to aggregate short texts to the pseudo-documents and uncover the cross-document word co-occurrence [8, 21, 24, 30]. However, the topics discovered by these models may be biased by the pseudo-documents generated heuristically. More specifically, many irrelevant short texts may be aggregated into the same pseudo-document.

Another strategy is to use the internal semantic relationships of the words to overcome the problem of lacking word co-occurrence. This strategy is proposed due to the fact that the semantic information of words has been effectively captured by the deep-neural-network-based word embedding techniques, such as word2vec [18] and Glove [20]. Several attempts [17, 22, 25] have been made to discover topics for short texts by leveraging semantic information

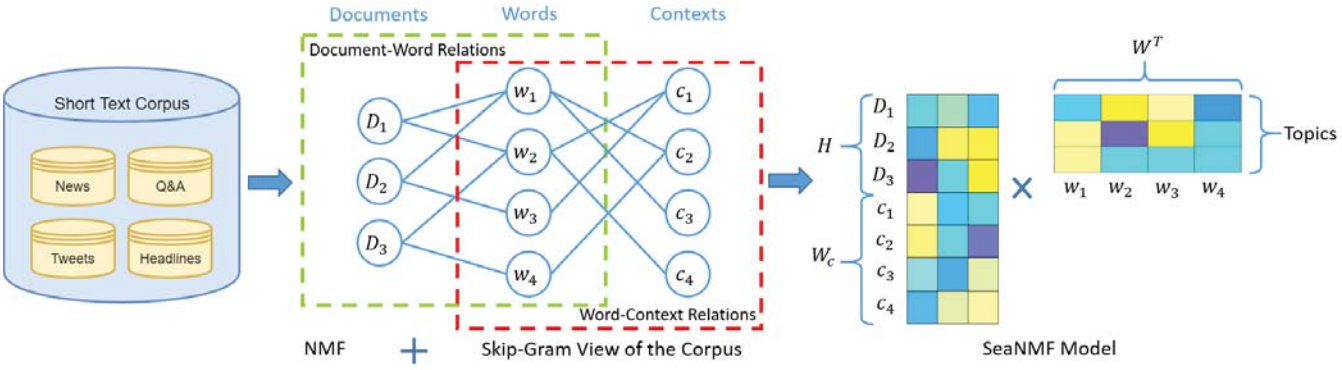


Figure 1: The overview of the proposed SeaNMF model for learning topics from the short text corpus, which is represented by a bi-relational matrix with both word-document and word-context correlations.

of the words from the existing sources, such as the word embeddings based on GoogleNews[1] and WiKipedia[2]. However, since there are many differences between the Wikipedia articles and the short texts, such word semantic representations may introduce the noise and bias to the topics.

Generally speaking, the word embedding can be useful for short text topic modeling because the words with similar semantic attributes are projected into the same region in the continuous vector space which will improve the clustering performance of the topic models. However, we find another way to boost the performance of the topic models using the skip-gram model with the negative sampling (SGNS). It is well known that SGNS can successfully capture the relationships between a word and its context in a small sliding window [18, 19]. Interestingly, for a short text corpus, each document can naturally be selected as a window. Therefore, the word-context semantic correlations will be effectively captured by SGNS. These correlations can be viewed as an alternative form of the word co-occurrence. It potentially overcomes the problem that arises due to the data sparsity.

There are a few recent studies which show that the SGNS algorithm is equivalent to factorizing a term correlation matrix [15, 16]. Thus, we raise some natural questions: **1)** Can we convert the matrix factorization problem to a non-negative matrix factorization problem? **2)** Can we incorporate this result into the conventional NMF for term-document matrix? **3)** Will the proposed model perform well on discovering topics for short texts? Motivated by these questions, we propose a novel semantics-assisted NMF (SeaNMF) model for short-text topic modeling which is outlined in Fig. 1. In this figure, the documents, words and contexts are denoted as D_i, w_i and c_i, respectively. The proposed SeaNMF model can capture the semantics from the short text corpus based on word-document and word-context correlations, and our objective function combines the advantages of both the NMF model for topic modeling and the skip-gram model for capturing word-context semantic correlations. In the figure, H, W_c and W are the vector representations of documents, contexts and words in the latent space. Each column of W represents a topic. We use a block coordinate descent algorithm to solve the optimizations. To achieve better interpretability, we also introduce a sparse version of the SeaNMF model.

The proposed models are compared with the other state-of-the-art methods on four real-world short text datasets. The quantitative experiments demonstrate the superiority of our models over several other existing methods in terms of topic coherence and document classification accuracy. The stability and consistency of SeaNMF are testified by parameter sensitivity analysis. Finally, we design an experiment to investigate the interpretability of the SeaNMF model. By visualizing the top keywords of different topics and analyzing their networks, we demonstrate that the topics discovered by SeaNMF are meaningful and their representative keywords are more semantically correlated. Hence, the proposed SeaNMF is an effective topic model for short texts.

The rest of this paper is organized as follows. In Section 2, we present related work. In Section 3, we propose the SeaNMF model and explain the optimization method used for learning the model. In Section 4, we introduce the datasets, comparison methods and evaluation metrics, as well as analyze the experimental results. Finally, we conclude our work in Section 5.

2 RELATED WORK

Topic modeling for short texts is a challenging research area and many models have been proposed to overcome the lack of contextual information. Most of the current studies are based on the generative probabilistic model, i.e., LDA [1]. Basically, there are three strategies to tackle the problem. The first strategy can capture the cross-document word co-occurrence via aggregating the short texts to the pseudo-documents. To aggregate the documents, some studies leverage the rich auxiliary contextual information, like authors, time, locations, etc. [8, 24]. For example, in [8], tweets posted by the same user are aggregated to a pseudo-document. However, this method cannot be applied to the corpus without auxiliary information. To overcome this disadvantage, another aggregation method is proposed, where the so-called latent pseudo-document is generated using the short texts according to their own topics [21, 30].

The second strategy considers to the word semantic information from a external corpus, like Wikipedia and Google news [17, 22,

[1]https://github.com/mmihaltz/word2vec-GoogleNews-vectors
[2]http://nlp.stanford.edu/projects/glove/

25]. It benefits a lot from the recently developed word embedding approaches based on neural networks [18, 19], which are efficient in uncovering the syntactic and semantic information of the words. For example, Xun et al. [25] train the word embeddings upon Wikipedia and use the semantic information as supplementary sources for their topic model. The third strategy directly makes use of word co-occurrence patterns in documents, i.e., short texts. It is also known as the Biterm model [26], since word-pairs co-occurring in the same short text are extracted during the topic modeling. All the above strategies have been demonstrated to be useful in discovering topics for short texts.

Although the NMF based methods have been successfully applied to topic modeling [2, 3, 9], very few of them are designed to discover topics for the short texts. In [27], Yan et al. propose a NMF model to learn topics for short texts by directly factorizing a symmetric term correlation matrix. However, since they formulate a quartic non-convex loss function, the algorithm proposed in the work is not reliable and stable. The recently proposed SymNMF [12, 13] can overcome this problem. However, it does not provide any good intuition for topic modeling. In addition, we cannot get the document representation from SymNMF directly. Therefore, the proposed method in this paper is the first work that considers to build a standard NMF-based topic model for the short texts.

3 PROPOSED METHOD

In this section, we will first provide some preliminaries along with the block coordinate descent method and its applications in NMF for topic modeling. Then, we will propose our SeaNMF model, and a block-coordinate descent algorithm to estimate latent representations of terms and short documents.

3.1 Notations

The frequently used notations in this section are summarized in Table 1.

Table 1: Notations used in this paper.

Name	Description
A	Term-document (word-document) matrix.
S	Word-context (semantic) correlation matrix.
W	Latent factor matrix of words.
W_c	Latent factor matrix of contexts.
H	Latent factor matrix of documents.
$\vec{w}_j$	Vector representation of word w_j.
$\vec{c}_j$	Vector representation of context c_j.
$\mathbb{R}_+$	Non-negative real numbers.
N	Number of documents in the corpus.
M	Number of distinct words in the vocabulary.

3.2 Preliminaries

3.2.1 **NMF for Topic Modeling.** The NMF method has been successfully applied to topic modeling, due to its superior performance in clustering high-dimensional data [2, 3, 11]. Given a corpus with N documents and M distinct words/terms/keywords in the vocabulary $\mathbb{V}$, we can use a term-document matrix $A \in \mathbb{R}_+^{M\times N}$ to represent it, where $\mathbb{R}_+$ denotes non-negative real numbers. Each column vector $A_{(:,j)} \in \mathbb{R}_+^{M\times 1}$ corresponds to a bag-of-word representation of document j in terms of M keywords. The term-document matrix can be approximated by two lower-rank matrices $W \in \mathbb{R}_+^{M\times K}$ and $H \in \mathbb{R}_+^{N\times K}$, i.e., $A \approx WH^T$, where $K \ll \min(M, N)$ is the number of latent factors (i.e., topics). Usually, this approximation can be formulated as follows:

$$\min_{W,H\geq 0} \|A - WH^T\|_F^2. \tag{1}$$

In topic models, the column vector $W_{(:,k)} \in \mathbb{R}_+^{M\times 1}$ represents the k-th topic in terms of M keywords, and its elements are the weights of the corresponding keywords. The row vector $H_{(j,:)} \in \mathbb{R}_+^{1\times K}$ is the latent representation for document j in terms of K topics. Similarly, we can view the row vector $W_{(i,:)} \in \mathbb{R}_+^{1\times K}$ as the latent semantic representation of word i. It is worth mentioning that there are many other divergences, which can be found in [4].

3.2.2 **Problem Statement.** Due to the data sparsity, the short texts are too short for the conventional topic models to effectively capture document-level word co-occurrence, which leads to the poor performance in topic learning. To tackle this problem, we first investigate the algorithms for estimating the factor matrices in NMF. For example, in the block coordinate descent (BCD) algorithm [10], the updating rules for W and H are shown as follows:

- **Update** W.

$$W_{(:,k)} \leftarrow \left[W_{(:,k)} + \frac{(AH)_{(:,k)} - (WH^TH)_{(:,k)}}{(H^TH)_{(k,k)}} \right]_+ \tag{2}$$

- **Update** H.

$$H_{(:,k)} \leftarrow \left[H_{(:,k)} + \frac{(A^TW)_{(:,k)} - (HW^TW)_{(:,k)}}{(W^TW)_{(k,k)}} \right]_+ \tag{3}$$

where $[x]_+ = \max(x, 0), \forall x \in \mathbb{R}$.

From the algorithm, we observe that the following lemma holds.

Lemma 3.1. *For the BCD algorithm, within each iteration:*

(1) *The keyword-vector $W_{(i,:)}^{t+1}$ is independent of vector $W_{(j,:)}^{t}$, when $1 \leq j \neq i \leq M$.*
(2) *The document-vector $H_{(i,:)}^{t+1}$ is independent of vector $H_{(j,:)}^{t}$, when $1 \leq j \neq i \leq N$.*

where t represents the t-th iteration.

Proof. To prove that $W_{(i,:)}^{t+1}$ is independent of $W_{(j,:)}^{t}$, $\forall j \neq i$, we only need to prove that $(WH^TH)_{(i,k)}$ is independent of $W_{(j,:)}$, $\forall 1 \leq k \leq K$. To simplify the proof, we use a symmetric matrix $B \in \mathbb{R}_+^{K\times K}$ to represent H^TH. Thus, we get $(WH^TH)_{(i,k)} = (WB)_{(i,k)} = W_{(i,:)} \cdot B_{(:,k)}$ which only depends on $W_{(i,:)}$. Hence, $W_{(i,:)}^{t+1}$ is independent of $W_{(j,:)}^{t}$,$\forall j \neq i$. Similarly, we can also prove that $H_{(i,:)}^{t+1}$ is independent of $H_{(j,:)}^{t}$. □

We also have the same conclusion for the gradient descent (GD) algorithm. Generally speaking, the relationship between different keywords strongly depends on the documents and vice-versa (see Fig. 1). However, due to the data sparsity, i.e., each document has only several keywords, the relationships of keywords are biased by

a lot of unrelated documents which results in poor clustering performance. Moreover, the relationships between the keywords and their contexts, i.e., semantic relationships, are not directly discovered by the BCD or GD algorithms in NMF. Therefore, a standard NMF model cannot effectively capture the word co-occurrence for short texts. In this paper, we will overcome this drawback by introducing additional dependence of the keywords on their contexts via neural word embedding (see Fig. 1).

3.2.3 **Neural Word Embedding.** Word embedding has been demonstrated to be an effective tool in capturing semantic relationships of the words. Represented by dense vectors, words with similar semantic and syntatic attributes can be found in the same area in the continuous vector space. One of the most successful word embedding methods is proposed by Mikolov et al. [18, 19], known as Skip-Gram with Negative-Sampling (SGNS). The objective function of SGNS is expressed as:

$$\log \sigma(\vec{w} \cdot \vec{c}) + \kappa \cdot \mathbb{E}_{c_{neg} \sim p(c)}[\log \sigma(-\vec{w} \cdot \vec{c}_{neg})], \tag{4}$$

where w and c represent word and one of its contexts in a sliding window, respectively. $\vec{w} \in \mathbb{R}^K$ and $\vec{c} \in \mathbb{R}^K$ are vector representations of them. $\sigma(\vec{w} \cdot \vec{c}) = 1/(1 + e^{-\vec{w} \cdot \vec{c}})$. c_{neg} is the sampled contexts, known as negative samples, drawn based on a unigram distribution $p(c)$. κ is the number of negative samples.

Recently, Levy et al. [15] have proven that SGNS is equivalent to factorizing a (shifted) word correlation matrix:

$$\vec{w} \cdot \vec{c} = \log \left(\frac{\#(w, c) \cdot \mathcal{D}}{\#(w) \cdot \#(c)} \right) - \log \kappa \tag{5}$$

where $\#(w, c)$ denotes the number of (w, c) pairs in a corpus. The total number of word-context pairs is $\mathcal{D} = \sum_{w,c \in \mathbb{V}} \#(w, c)$. Similarly, $\#(w) = \sum_{c \in V} \#(w, c)$ and $\#(c) = \sum_{w \in \mathbb{V}} \#(w, c)$ represent the number of times w and c occur in all possible word-context pairs, respectively. $p(c)$ in Eq. (4) is expressed as $p(c) = \#(c)/\mathcal{D}$. It is worth mentioning that the $\log((\#(w, c) \cdot \mathcal{D})/(\#(w) \cdot \#(c)))$ is known as the pointwise mutual information (PMI). Therefore, based on this concern, an alternative word representation method was proposed in [15], where the positive constraint is applied to the PMI matrix (PPMI), and then it is factorized by a singular value decomposition method. The Eq. (5) reveals the internal relationships between the word and its context, which is critical to overcome the problem of lacking word co-occurrence. In this paper, we will leverage the word-context semantic relationships to boost the performance of our models.

3.3 The SeaNMF Model

In this section, we propose a novel semantics-assisted NMF (SeaNMF) model to learn topics from the short texts. Our model incorporates the semantic information using the word embeddings into the model training, which enable SeaNMF to recover word co-occurrence from semantic relationships between keywords and their contexts (see Fig. 1).

3.3.1 Model Formulation. One challenge of our work is to appropriately introduce the word semantics to NMF. Since the latent matrix $W \in \mathbb{R}_+^{M \times K}$ (The elements of W are non-negative), we apply the non-negative constraints on both word and context vectors. Therefore, $\vec{w} \in \mathbb{R}_+^K$ and $\vec{c} \in \mathbb{R}_+^K$ hold. Given a keyword $w_i \in \mathbb{V}$, we set $W_{(i,:)} = \vec{w}_i$. To reveal the semantic relationships between the keywords and their context, a matrix W_c is defined for the words in contexts. Thus, $W_c(j, :) = \vec{c}_j$ for $c_j \in \mathbb{V}$.

With the word and context representations, we can define a semantic (word-context) correlation matrix S which reveals relationships between the keyword and their contexts. Hence, we have

$$S \approx W W_c^T. \tag{6}$$

The matrix S can be obtained from the skip-gram view of the corpus. Here, we define each element S_{ij} as follows:

$$S_{ij} = \left[\log \left(\frac{\#(w_i, c_j)}{\#(w_i) \cdot p(c_j)} \right) - \log \kappa \right]_+, \tag{7}$$

where $p(c_j)$ is a unigram distribution for sampling a context c_j. Different from Eq. (5), it is defined as

$$p(c_j) = \frac{\#(c_j)^\gamma}{\sum_{c_j \in \mathbb{V}} \#(c_j)^\gamma}, \tag{8}$$

where γ is a smoothing factor. It should be noted that S need not necessarily be symmetric. Specifying the sliding windows is a critical component of the skip-gram model. However, for the short texts, this work turns out to be simple. That is, we can naturally view each short document as a window, since each window will have only a few words. Therefore, the total number of windows is equal to the number of documents. Finally, $\#(w_i, c_j)$, $\#(w_i)$, $\#(c_j)$ and $\mathcal{D}$ will be calculated accordingly.

REMARK 1. *The semantic correlation matrix S is not required to be symmetric.*

REMARK 2. *In this paper, each short text is viewed as a window. Therefore, the size of each window in the skip-gram model is equal the length of the corresponding short text. The total number of windows is equal to the number of short texts.*

With the term-document matrix and the semantic correlation matrix, the objective function is expressed as follows:

$$\min_{W, W_c, H \geq 0} \left\| \begin{pmatrix} A^T \\ \sqrt{\alpha} S^T \end{pmatrix} - \begin{pmatrix} H \\ \sqrt{\alpha} W_c \end{pmatrix} W^T \right\|_F^2 + \psi(W, W_c, H), \tag{9}$$

where $\alpha \in \mathbb{R}_+$ is a scale parameter. $\psi(W, W_c, H)$ is a penalty function for SeaNMF, which will be specified for a different purpose, such as the sparsity. In this paper, we will primarily demonstrate that SeaNMF is an effective topic model for the short texts.

3.3.2 Optimization. Suppose $\psi(W, W_c, H) = 0$, a block coordinate descent (BCD) algorithm can be used to solve Eq. (9). We take the derivatives of the objective function with respect to the vectors $W_{(:,k)}$, $W_{c(:,k)}$ and $H_{(:,k)}$. By setting them to zero, we get the updating rules as follows:

- **Update** W

$$W_{(:,k)} \leftarrow [W_{(:,k)} + \frac{(AH)_{(:,k)} + \alpha(SW_c)_{(:,k)} - (WH^TH)_{(:,k)} - \alpha(WW_c^TW_c)_{(:,k)}}{(H^TH)_{(k,k)} + \alpha(W_c^TW_c)_{(k,k)}}]_+ \tag{10}$$

- **Update** W_c

$$W_{c(:,k)} \leftarrow \left[W_{c(:,k)} + \frac{(SW)_{(:,k)} - (W_c W^T W)_{(:,k)}}{(W^TW)_{(k,k)}} \right]_+ \tag{11}$$

From lemma 3.1, the document representation H is independent of W_c and S, therefore, the update rule for H is the same as Eq. (3).

Algorithm 1: The SeaNMF Algorithm

Input: Term-document matrix A;
Semantic correlation matrix S;
Number of topics K, α;
Output: W, W_c, H;

Initialize: $W \geq 0$, $W_c \geq 0$, $H \geq 0$ random real numbers;
$t = 1$;
repeat
  for *k=1,K* **do**
    Compute $W^t_{(:,k)}$ by Eq. (10);
    Compute $W^t_{c(:,k)}$ by Eq. (11);
    Compute $H^t_{(:,k)}$ by Eq. (3);
  end
  $t = t + 1$;
until *Converge*;

The BCD algorithm for SeaNMF is summarized in Algorithm 1. We first build the term-document matrix A using the bag-of-word representation. Then, we calculate the semantic correlation matrix S by Eq. (7). The latent factor matrices W, W_c and H are initialized randomly with non-negative real numbers. Then, within each iteration, their coordinates will be updated column-wise. After each update, $W_{(:,k)}$ and $W_{c(:,k)}$ will be normalized to have a unit ℓ_2-norm. We will repeat this iteration until the algorithm converges.

3.3.3 Intuitive Explanation. We further demonstrate that Eq. (10) is equivalent to the following three updating procedures.

$$W^1_{(:,k)} \leftarrow W_{(:,k)} + \frac{(AH)_{(:,k)} - (WH^TH)_{(:,k)}}{(H^TH)_{(k,k)}} \tag{12}$$

$$W^2_{(:,k)} \leftarrow W_{(:,k)} + \frac{(SW_c)_{(:,k)} - (WW_c^TW_c)_{(:,k)}}{(W_c^TW_c)_{(k,k)}} \tag{13}$$

$$W_{(:,k)} \leftarrow \left[\lambda W^1_{(:,k)} + (1-\lambda)W^2_{(:,k)}\right]_+ \tag{14}$$

where $\lambda = \frac{(H^TH)_{(k,k)}}{(H^TH)_{(k,k)}+\alpha(W_c^TW_c)_{(k,k)}} \in [0,1]$. □

As we can see, Eq. (12) is the same as Eq. (2) for the standard NMF. It tries to project the words in the same documents into the same region of the space using the term-document matrix. On the other hand, the Eq. (13) tries to move the words close to each other if they share the common context keywords. Therefore, it increases the coherence of the topics. For example, in Fig. 1, w_1 and w_4 do not appear in the same document. However, since they both have w_2 as context keyword, they may be semantically correlated. Take two short texts "iphone ios system" and "galaxy android system" as an example. "iphone" and "ios" do not appear in the second sentence, and "galaxy" and "android" do not appear in the first sentence. Thus, the correlations between "iphone, ios" and "galaxy, android" are minor in the standard NMF. However, in SeaNMF, the correlations are enhanced by Eq. (13) using the fact that they share the common keywords "system". The overall updating procedure, given in Eq. (14), is a linear combination of Eq. (12) and (13) which guarantees the top keywords in each topic are highly correlated.

3.3.4 Computational Complexity. We have noticed that the proposed SeaNMF model maintains the same formation (Eq. (9)) as that of the standard NMF (Eq. (1)), therefore, its computational complexity is $O((M+N)MK)$ within a single iteration of updating factor matrices. Since for short text corpus, the number of keywords is usually less than the number of documents, i.e, $M < N$, we have $M + N < 2N$. Therefore, the computational complexity of SeaNMF for short texts is reduced to $O(NMK)$, which is the same as that of standard NMF [10]. However, due to the data sparsity for short texts, this complexity can be further reduced. In details, it can be seen from Eqs. (10), (11) and (3), the complexity is dominated by the calculations of AH, SW_c, A^TW. Without considering the sparsity, their computational costs are $O(MNK)$, $O(MMK)$, $O(NMK)$, respectively. However, since A and S are sparse matrices, which can be seen in Table 2, we only need to multiply the non-zero elements with factor matrices. Suppose the numbers of non-zero elements in A and S are z_A and z_S, the complexity of calculating AH, SW_c, A^TW will be $O(z_AK)$, $O(z_SK)$, $O(z_AK)$, respectively. Therefore, the proposed SeaNMF model has the complexity of $O(\max(z_A, z_S)K)$, where $\max(z_A, z_S) \ll NM$ and $K \ll \min(N, M)$, which is much cheaper than the standard NMF.

3.4 The Sparse SeaNMF Model

In standard topic models, words are represented by dense vectors in a continuous real space. Specifically, in SeaNMF, we use the low-rank factor matrix W to encode the words. Introducing sparsity to W will reduce the active components of the word vectors, which will make it easy to interpret the topics.

Considering a better interpretability of the model, we introduce the Sparse SeaNMF (SSeaNMF) model, where we apply the sparsity constraint to W and express the penalty function as follows:

$$\psi(W, W_c, H) = \beta\|W\|_1^2, \tag{15}$$

where $\|\cdot\|_1$ represents the ℓ_1-norm. Since the sparsity is only applied to W, the BCD algorithm for updating W is modified to

$$W_{(:,k)} \leftarrow [W_{(:,k)} + \frac{(AH)_{(:,k)} + \alpha(SW_c)_{(:,k)} - (WH^TH)_{(:,k)} - \alpha(WW_c^TW_c)_{(:,k)} + \beta \cdot 1_K}{(H^TH)_{(k,k)} + \alpha(W_c^TW_c)_{(k,k)} + \beta}]_+ \tag{16}$$

where $1_K \in \mathbb{R}^{M\times 1}$ and $1_{K(i,:)} = -\sum_{k=1}^{K} W_{(i,k)}, \forall 1 \leq i \leq M$.

Updating procedures for W_c and H remain the same as in Eq. (11) and Eq. (3), respectively. Compared with standard SeaNMF, calculating 1_K will not significantly increase the computational complexity of the algorithm.

4 EXPERIMENTAL RESULTS

In this section, we will demonstrate the promising performance of our models by conducting extensive experiments on different real-world datasets. We will introduce the datasets, evaluation metrics and baseline methods, and then explain different sets of results.

4.1 Datasets Used

Our experiments are carried out on four real-world short text datasets corresponding to four types of applications, i.e., News, Questions&Answers, Microblogs and Article headlines.

• **Tag.News**. This data set is a part of the TagMyNews dataset[3], which is composed of news, snippets and tweets. After removing the stopwords, we only keep the news with at most 25 keywords. The articles in the dataset belong to one of the following 7 categories: Business, Entertainment, Health, Sci&Tech, Sport, US and World.

• **Yahoo.Ans**. This dataset is a subset extracted from the Yahoo! Answers Manner Questions, version 2.0[4]. In our dataset, we collect the subjects of the Questions from 10 different categories, including Financial Service, Diet&Fitness, etc.

• **Tweets**. The original Tweets dataset is collected and labeled by Zubiaga et al. [29]. We select 15 different categories from the dataset, i.e., Arts, Business, Computers, Games, Health, Home, News, Recreation, Reference, Regional, Science, Shopping, Society, Sports and World. For each category, we sample 2500~3000 distinct tweets with at least two keywords.

• **DBLP**. The raw DBLP dataset is available at [5]. In our dataset, we collect the titles of the conference papers from the following 4 categories: Machine Learning, Data Mining, Information Retrieval and Database.

• **GoogleNews(300d)**. This dataset is obtained from [6]. It contains 3 million English words which are embedded into 300 dimensional latent space by performing the word2vec model [19] on Google News corpus which consists of 3 billion running words. It is used to train the comparison method GPUDMM [17]

Table 2: Basic statistics of the datasets used in this paper.

Data Set	#docs	#terms	density(A)	density(S)	doc-length	#cats
Tag.News	28658	11525	1.2861%	0.1369%	18.14	7
Yahoo.Ans	40754	4334	0.1997%	0.0973%	4.30	10
Tweets	43413	10279	0.2744%	0.0713%	7.73	15
DBLP	15001	2447	0.7693%	0.2677%	6.64	4
Yahoo.CA	30686	4334	5.0532%	0.7754%	42.61	-
ACM.IS	36392	2447	4.2667%	1.9494%	77.49	-

Some basic statistics of these datasets are shown in Table 2. In this table, '#docs' represents the number of documents in each dataset. '#terms' is the number of keywords in the vocabulary. 'density' is defined as $\frac{\#\text{non-zero}}{\#\text{docs}\cdot\#\text{terms}}$, where #non-zero is the number of non-zero elements in the matrix. The 'density(A)' and 'density(S)' represent the density of term-document matrix (A) and semantic correlation matrix (S), respectively. 'doc-length' represents the average length of the documents. '#cats' denotes the number of distinct categories.

In our experiments, we also leverage the following two datasets as external sources in the evaluations. It should be noted that they are NOT used to train the models.

• **Yahoo.CA**. From the Yahoo! Answers Manner Questions, version 2.0, we collect the content and best answer for each question, and construct a new regular-sized document sets, namely, Yahoo.CA.

• **ACM.IS**. This dataset is part of ACM IS abstract dataset[7], which contains the abstracts of ACM information system papers published between 2002 and 2011.

4.2 Evaluation Metrics

In this paper, we will use the topic coherence and document classification accuracy for our evaluation.

• **Topic Coherence**. Given a topic k, the PMI score is calculated by the following equation:

$$C_k = \frac{2}{\mathcal{N}(\mathcal{N}-1)} \sum_{1 \le i < j \le \mathcal{N}} \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{17}$$

where $\mathcal{N}$ is the number of most probable words in this topic. $p(w_i, w_j) = \#(w_i, w_j)/\mathcal{D}$ is the probability of the words w_i and w_j co-occurring in the same document. $p(w_i) = \#(w_i)/\mathcal{D}$ and $p(w_j) = \#(w_j)/\mathcal{D}$ are the marginal probabilities. The average PMI score over all the topics will be used to evaluate the quality of the topic models. However, Quan et al. [21] have shown that the average PMI score, that works well for regular-sized documents, is still problematic for short texts, which means a gold-standard topic may be assigned with a low PMI score.

In this paper, we leverage the following strategy to overcome this problem. First, we calculate the PMI score based on the four short text datasets as usual. Second, for Yahoo.Ans and DBLP datasets, we calculate the PMI score based on the external corpus, i.e., Yahoo.CA and ACM.IS, which are composed of regular documents. The results in both experiments will be used to demonstrate the effectiveness of our models. We emphasize that Yahoo.CA and ACM.IS do not participate in the training of our models.

In our experiments, we set $\mathcal{N} = 10$. It also should be noted that the difference between Eq. (17) and the PMI score used in [30] is that we do not consider the co-occurrence of the same word.

• **Document Classification**. Another popular way to evaluate the effectiveness of the topic models is to leverage the latent document representations for external tasks. In our experiments, we will conduct short text classification on all the datasets, whose documents have been labeled. A five-fold cross validation is used to evaluate the performance of the classification, where each corpus is randomly split into training and testing sets with a ratio of $4:1$. Then, the documents are classified by LIBLINEAR package [8] [6].

Finally, the quality of the classification is measured by averaged precision, recall and F-score.

4.3 Comparison Methods

We compare the performance of our models with the following state-of-the-art methods.

• **Latent Dirichlet Allocation (LDA)**. LDA [1] is a well-known baseline method in the topic modeling which performs well on the regular-sized documents. In this paper, we use a Python implementation[9] of LDA with a collapsed Gibbs sampling.

• **Non-negative Matrix Factorization (NMF)**. NMF [10] is an unsupervised method that can perform dimension reduction and clustering simultaneously. It has found applications in a range of

[3] http://acube.di.unipi.it/datasets/
[4] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l
[5] http://dblp.uni-trier.de/
[6] https://github.com/mmihaltz/word2vec-GoogleNews-vectors
[7] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/27695
[8] https://www.csie.ntu.edu.tw/~cjlin/liblinear/
[9] https://github.com/shuyo/iir/tree/master/lda

areas, including topic modeling. In our experiments, the NMF[10] is implemented in Python with a block coordinate descent algorithm.

• **Pseudo-document-based Topic Model (PTM)**. PTM [30] introduces *pseudo-documents* into the topic model, which implicitly aggregates short texts without auxiliary information. It is one of the most recent methods for discovering topics from the short text corpus.

• **GPUDMM**. The GPUDMM [17] for short-text topic modeling is based on the Dirichlet Multinomial Mixture model. During the sampling process using the generalized Pólya urn model, it promotes the semantically related words in each topic by leveraging the external word semantic knowledge, i.e., word vectors, from very large corpus. In this paper, we will use the GoogleNews(300d) dataset as the external resource.

In our experiments, the default number of topics is set to $K = 100$. For LDA, we set parameters $\alpha = 0.1$ and $\beta = 0.01$, since the weak prior can give a better performance for short texts [30]. For PTM and GPUDMM, we use the default hyper-parameter settings. In details, we set parameters $\alpha = 0.1$, $\lambda = 0.1$ and $\beta = 0.01$ for PTM. For GPUDMM, we set parameters $\beta = 0.1$. In LDA, PTM and GPUDMM, Gibbs sampling is run for 2000 iterations. For SeaNMF, we set $\alpha = 1.0$ for Tag.News and Tweets and $\alpha = 0.1$ for Yahoo.Ans and DBLP. To calculate S, we set $\kappa = 1.0$ and $\gamma = 1.0$. In SSeaNMF, we set $\beta = 0.1$. We also set the seed for the random number generator to 0 for NMF, SeaNMF and SSeaNMF to make sure the results are consistent and independent of random initial states. The codes for SeaNMF has been publicly available at [11].

4.4 Results

4.4.1 Topic Coherence Results. We first present the topic coherence results of our models and other comparison methods in Tables 3 and 4. We use the bold font to show the best performance values and the underline to highlight the second best values.

Table 3: Topic coherence results in terms of PMI.

	Tag.News	Yahoo.Ans	Tweets	DBLP
LDA	1.5048	1.2957	1.1637	0.9346
NMF	1.6414	1.1394	1.8045	0.9184
PTM	1.6628	1.1311	1.3745	0.8505
GPUDMM	0.9751	0.5798	0.9213	0.2815
SeaNMF	**3.6318**	**1.7553**	4.1477	1.6137
SSeaNMF	3.6053	1.6081	**4.1979**	**1.6239**

From Table 3, we observe that our models outperform the standard NMF, which indicates that SeaNMF is effective for learning topics from short texts. Compared with LDA and recent PTM, SeaNMF shows significant improvements, which implies that our models discover more coherent topics. To better understand the poor performance of GPUDMM in all cases, we visualize the top keywords in each topic, where we find that many top keywords (e.g. 'extraction', 'extracting' and 'extract') are semantically correlated, but they do not tend to appear in the same document. Another possible reason is that the word semantic relationships in Google News and other datasets are different, so that the general semantics knowledge from Google News may not work well on discovering topics from these datasets.

Table 4: Topic coherence results with Yahoo.CA and ACM.IS.

	Yahoo.Ans/Yahoo.CA	DBLP/ACM.IS
LDA	0.6540	0.4282
NMF	0.5261	0.3626
PTM	0.6504	0.4431
GPUDMM	0.3302	-0.0159
SeaNMF	**1.1094**	0.6641
SSeaNMF	1.0188	**0.6447**

As discussed in topic coherence section, since the PMI scores are problematic for short texts, we also evaluate topic coherence based on external corpus which are composed of long documents. After training different models on Yahoo.Ans, we extract the top keywords from each topic, and then calculate the PMI scores based on the Yahoo.CA corpus. Similarly, for DBLP, the PMI scores are calculated based on ACM.IS dataset. The results obtained on these external corpus are presented in Table 4. From the table, we find that SeaNMF outperforms the other baseline methods. Therefore, from our topic coherence results, we demonstrate that by leveraging the word semantic correlations, SeaNMF can capture more coherent topics from short texts.

4.4.2 Document Classification Results. In addition to the topic coherence, we also compared the document classification performance of different methods. As we can see from Table 5, both the best and the second best results are achieved by our models on Tag.News, Yahoo.Ans and Tweets. This demonstrates that our models are effective in the document classification for short texts. Compared with the conventional topic models, such as LDA and NMF, SeaNMF has a significant improvement in terms of different classification measures. The SeaNMF models also perform better than PTM, which attempts to capture the cross-document word correlations by aggregating similar short texts into pseudo documents. This comparison demonstrates that the word correlations obtained from skip-gram view of the corpus play an important role in capturing high quality semantics, given the performance of standard NMF is not as good as that of LDA. In Table 5, we also observe that the GPUDMM model performs better than the other baseline methods. The difference between GPUDMM and SeaNMF is that GPUDMM explicitly makes use of the term correlations obtained from the pre-trained word representations on the external large corpus, while SeaNMF is only based on the short text corpus itself. Thus, given an external resource, like Google News, the performance of GPUDMM cannot be guaranteed across different short texts. In summary, the classification results have shown that SeaNMF is a superior topic model for short texts, even without using the auxiliary information or external sources, or aggregating the short texts.

It should be noted that the results based on the Tweets dataset are more reliable because the number of tweets in different categories is almost the same, which avoids the problems caused by the so-called 'imbalanced classes'. As we can see in Tables 5, SeaNMF has on an average more than 12% improvements over the other baseline methods with respect to precision, recall, F-score.

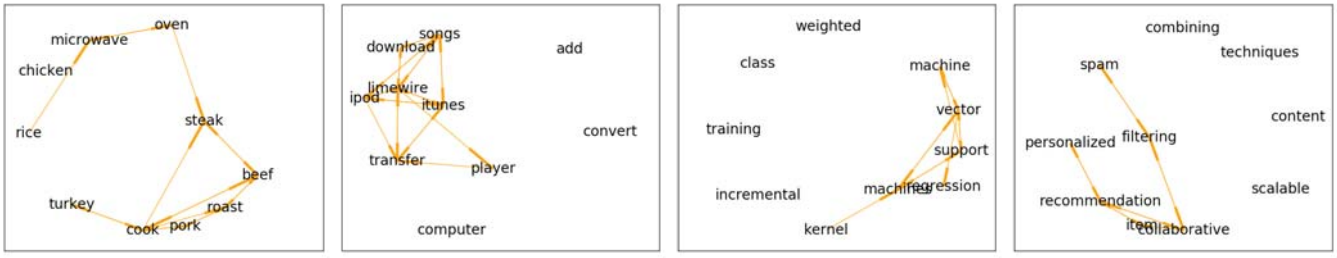
[10] https://github.com/kimjingu/nonnegfac-python
[11] https://github.com/Text-Analytics/SeaNMF

Table 5: Performance comparison of various methods on document classification.

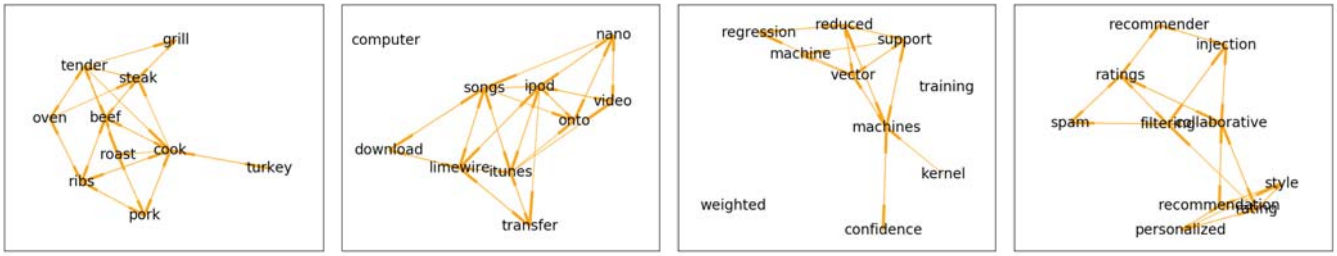
	Tag.News			Yahoo.Ans			Tweets			DBLP		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
LDA	0.7323	0.7184	0.7239	0.5929	0.5738	0.5659	0.3827	0.3867	0.3758	0.6081	0.5973	0.5994
NMF	0.6763	0.6371	0.6507	0.6303	0.5470	0.5706	0.3677	0.3517	0.3506	0.6393	0.6226	0.6273
PTM	0.7525	0.7396	0.7444	0.6390	0.6038	0.6026	0.3941	0.3838	0.3786	0.6424	0.6367	0.6379
GPUDMM	0.7843	0.7712	0.7760	0.5954	0.6308	0.5995	0.3985	0.4066	0.3903	0.6670	0.6573	0.6586
SeaNMF	0.7868	0.7786	0.7821	0.6566	0.6338	0.6366	**0.4648**	0.4555	**0.4527**	0.6648	0.6552	0.6575
SSeaNMF	**0.7894**	**0.7801**	**0.7841**	**0.6603**	**0.6369**	**0.6401**	0.4592	**0.4568**	0.4516	**0.6700**	**0.6613**	**0.6636**

Table 6: Discovered topics by the proposed method. The word is colored in red if its degree is less than 2. The numbers in the parentheses represent the frequency of the word in the corpus. NMF-k corresponds to the k-th topic discovered by the NMF model.

	Yahoo.Ans				DBLP			
Category	Cooking and Recipes		Blues		Machine Learning		Data Mining	
	NMF-24	SeaNMF-47	NMF-54	SeaNMF-50	NMF-100	SeaNMF-45	NMF-72	SeaNMF-98
PMI	2.7291	3.1713	2.6674	3.3517	1.4570	1.7215	1.2636	1.9810
Top-10 keywords	cook(381)	cook(381)	songs(257)	songs(257)	support(228)	support(228)	filtering(147)	filtering(147)
	chicken(168)	roast(54)	ipod(143)	ipod(143)	vector(150)	vector(150)	collaborative(122)	collaborative(122)
	turkey(72)	oven(67)	download(179)	computer(216)	machines(95)	machines(95)	content(166)	recommendation(47)
	roast(54)	pork(40)	computer(216)	download(179)	machine(116)	machine(116)	scalable(130)	personalized(62)
	rice(80)	beef(56)	itunes(54)	transfer(75)	regression(127)	regression(127)	combining(118)	spam(37)
	oven(67)	grill(50)	player(94)	onto(51)	class(104)	kernel(151)	spam(37)	recommender(27)
	beef(56)	turkey(72)	limewire(70)	itunes(54)	training(79)	training(79)	recommendation(47)	injection(5)
	pork(40)	steak(50)	transfer(75)	limewire(70)	kernel(151)	confidence(19)	personalized(62)	style(15)
	steak(50)	tender(11)	add(138)	video(71)	incremental(105)	reduced(5)	item(29)	rating(8)
	microwave(51)	ribs(16)	convert(118)	nano(31)	weighted(67)	weighted(67)	techniques(115)	ratings(6)



(a) NMF-24 (Yahoo.Ans) (b) NMF-54 (Yahoo.Ans) (c) NMF-100 (DBLP) (d) NMF-72 (DBLP)



(e) SeaNMF-47 (Yahoo.Ans) (f) SeaNMF-50 (Yahoo.Ans) (g) SeaNMF-45 (DBLP) (h) SeaNMF-98 (DBLP)

Figure 2: Network Visualizations of the keywords obtained by the NMF and SeaNMF models on Yahoo.Ans and DBLP datasets.

4.5 Parameter Sensitivity

In this section, we will demonstrate the stability and consistency of SeaNMF by varying the parameters α, κ and γ.

The parameter α is the weight for factorizing the word semantic correlation matrix. Here, we study the effects of α on the topic coherence and classification accuracy on DBLP. It can be seen from Fig. 3 that the topic coherence increases rapidly as we increase the weight when $\alpha \in (0, 1]$. However, it stays almost constant after $\alpha > 1$. This clearly shows that SeaNMF is effective for short texts just because it leverages the word semantic correlations.

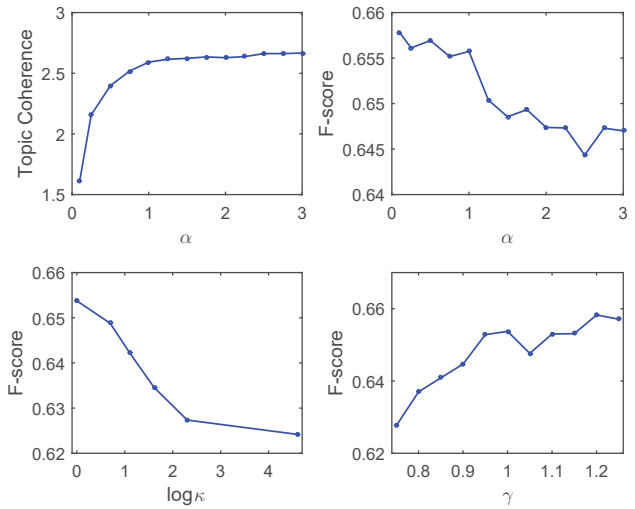


Figure 3: Topic coherence and classification performance by varying α, κ and γ.

We also observe that a better topic coherence does not imply better document classification performance. As we can see in Fig. 3, the F-score decreases as α increasing. Therefore, for a short text collection, a highly coherent topic is not the same as a high quality topic which is consistent with the findings of others in the literature [21]. We also notice that the F-score does not significantly change with α, i.e., the change is less than 0.02. Hence, SeaNMF is a stable topic model for short texts.

The parameters κ and α play an important role in constructing the semantic correlation matrix S. κ affects the sparsity of S. Large κ leads to very sparse S and sparse S implies that the words are less correlated. As shown in Fig. 3, the F-score is reduced when we increase κ. γ is a smoothing factor for the probability of sampling a context. From the figure, the F-score is slightly improved when γ is increased. To summarize, both parameters affect the quality of topics by changing the semantic correlation matrix. It implies that the word semantic correlations are critical to SeaNMF.

4.6 Semantic Analysis of Topics

In this section, we show that the topics discovered by SeaNMF are meaningful by visualizing the top keywords. They will be compared with the top keywords given by the standard NMF method.

After training the NMF model on Yahoo.Ans and DBLP datasets, we select the topics with high PMI scores. Then, we find the most similar topic obtained from SeaNMF for each of them based on the top keywords. The lists of the top keywords in the selected topics obtained are shown in Table 6. As we can see, two topics for Yahoo.Ans are about cooking and the technical problems on downloading or transferring songs. The two topics selected from DBLP are on publications related with machine learning and data mining.

To demonstrate the topics discovered by SeaNMF are more semantically correlated, we use the selected top keywords in each topic to construct the word networks. More specifically, suppose the top keyword list is denoted as $\{w_i\}_{i=1}^{10}$, we first find 30 most correlated words $\{v_j\}_{j=1}^{30}$ for each keyword w_{i_0} based on the positive PMI matrix. If a keyword $w_{i_1} \in \{w_i\} \cap \{v_j\}$, $i_1 \neq i_0$, we draw an edge from w_{i_0} to w_{i_1}.

As we can see from Fig. 2, all the graphs for the standard NMF model are very sparse. Some keywords with higher frequency in the corpus have lower degree which means that they are less correlated with the other words. For example, the frequency of 'chicken' is high, however, its most correlated words do not contain the other keywords and it is not in the most correlated word lists of the other keywords. In the standard topic modeling, these keywords might be viewed as noise. In Table 6, the keywords with degree less than two are colored in red. We can see that the topics obtained from the standard NMF model are noisy. On the other hand, we conduct the same experiments on our SeaNMF model. From Table 6 and Fig. 2, we can see that topics discovered by our SeaNMF model have less noisy words and the top keywords are more correlated. Therefore, these semantic analysis results demonstrate that the SeaNMF model can discover meaningful and consistent topics for short texts.

5 CONCLUSION

In this paper, we introduce a semantics-assisted NMF (SeaNMF) model to discover topics for the short texts. The proposed model leverages the word-context semantic correlations in the training, which potentially overcomes the problem of lacking context that arises due to the data sparsity. The semantic correlations between the words and their contexts are learned from the skip-gram view of the corpus, which was demonstrated to be effective for revealing word semantic relationships. We use a block coordinate descent algorithm to solve our SeaNMF model. To achieve a better model interpretability, a sparse SeaNMF model is also developed. We compared the performance of our models with several other state-of-the-art methods on four real-world short text datasets. The quantitative evaluations demonstrate that our models outperform other methods with respect to widely used metrics such as the topic coherence and document classification accuracy. The parameter sensitivity results demonstrate the stability and consistency of the performance of our SeaNMF model. The qualitative results show that the topics discovered by SeaNMF are meaningful and their top keywords are more semantically correlated. Hence, we conclude that the proposed SeaNMF is an effective topic model for short texts.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation grants IIS-1619028, IIS-1707498 and IIS-1646881, and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (No. NRF-2016R1C1B2015924).

REFERENCES

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[2] Jaegul Choo, Changhyun Lee, Chandan K Reddy, and Haesun Park. 2013. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE transactions on visualization and computer graphics* 19, 12 (2013), 1992–2001.

[3] Jaegul Choo, Changhyun Lee, Chandan K Reddy, and Haesun Park. 2015. Weakly supervised nonnegative matrix factorization for user-driven clustering. *Data Mining and Knowledge Discovery* 29, 6 (2015), 1598–1621.

[4] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. 2009. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons.

[5] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391–407.

[6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of machine learning research* 9, Aug (2008), 1871–1874.

[7] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 50–57.

[8] Liangjie Hong and Brian D Davison. 2010. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*. ACM, 80–88.

[9] Hannah Kim, Jaegul Choo, Jingu Kim, Chandan K Reddy, and Haesun Park. 2015. Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 567–576.

[10] Jingu Kim, Yunlong He, and Haesun Park. 2014. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization* 58, 2 (2014), 285–319.

[11] Da Kuang, Jaegul Choo, and Haesun Park. 2015. Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional Clustering Algorithms*. Springer, 215–243.

[12] Da Kuang, Chris Ding, and Haesun Park. 2012. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*. SIAM, 106–117.

[13] Da Kuang, Sangwoon Yun, and Haesun Park. 2015. SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization* 62, 3 (2015), 545–574.

[14] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

[15] Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2177–2185.

[16] Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3 (2015), 211–225.

[17] Chenliang Li, 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*. ACM, 165–174.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[19] Tomas Mikolov, 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*. 3111–3119.

[20] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[21] Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. 2015. Short and Sparse Text Topic Modeling via Self-aggregation. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, 2270–2276.

[22] Vivek Kumar Rangarajan Sridhar. 2015. Unsupervised topic modeling for short texts using distributed representations of words. In *Proceedings of NAACL-HLT*. 192–200.

[23] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 841–842.

[24] Zhongyuan Wang and Haixun Wang. 2016. Understanding Short Texts. In *ACL 2016 Tutorial*. ACL, 1–18.

[25] Guangxu Xun, Vishrawas Gopalakrishnan, Fenglong Ma, Yaliang Li, Jing Gao, and Aidong Zhang. 2016. Topic Discovery for Short Texts Using Word Embeddings. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 1299–1304.

[26] Xiaohui Yan, 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*. ACM, 1445–1456.

[27] Xiaohui Yan, Jiafeng Guo, Shenghua Liu, Xueqi Cheng, and Yanfeng Wang. 2013. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 749–757.

[28] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *European Conference on Information Retrieval*. Springer, 338–349.

[29] Arkaitz Zubiaga and Heng Ji. 2013. Harnessing web page directories for large-scale classification of tweets. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 225–226.

[30] Yuan Zuo, 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*. ACM, 2105–2114.