Combine Topic Modeling with Semantic Embedding: Embedding Enhanced Topic Model

Peng Zhang, Suge Wang, Deyu Li, Xiaoli Li, and Zhikang Xu

Abstract—Topic model and word embedding reflect two perspectives of text semantics. Topic model maps documents into topic distribution space by utilizing word collocation patterns within and across documents, while word embedding represents words within a continuous embedding space by exploiting the local word collocation patterns in context windows. Clearly, these two types of patterns are complementary. In this paper, we propose a novel integration framework to combine the two representation methods, where topic information can be transmitted into corresponding semantic embedding structure. Based on this framework, we construct a Embedding Enhanced Topic Model (EETM), which can improve topic modeling and generate topic embeddings by leveraging the word embedding. Extensive experimental results show that EETM can learn high-quality document representations for common text analysis tasks across multiple data sets, indicating it is very effective for merging topic models with word embeddings.

Index Terms—Topic model, Word embedding, Topical embedding, Representation learning.

1 Introduction

In many text analysis tasks, such as text categorization, sentiment analysis and text clustering, unstructured text data need to be first converted to structured forms or representations, namely known as *Text Representation Learning* [1], [2], [3], [4]. Clearly, the quality of text representation is essential for the subsequent text analysis tasks. As such, researchers have proposed some methods to address this problem, where topic models and word embeddings are two major representatives, aiming to produce high quality text representations.

Topic models discover the *topics* that occur in a corpus according to words' collocation patterns within and across documents [5], [6], [7]. As such patterns reflect their semantic relations, topic models can thus discover coherent topics that comprise semantically relevant words. Represented by Latent Dirichlet Allocation (LDA) [8], typical topic models can be regarded as document-level representation methods, which assume that each document has a discrete topic distribution. The probability of a topic in a certain document does not reflect what words are used in it, but indicates how many words from that topic present in the document. In addition, the word collocation patterns at document-level, similar to bag-of-words models, ignore the sequential relationship between words. Information implied in words' sequence, such as syntax and dependence of words, thus can not be modeled by LDA either.

Embedding methods, represented by word embeddings, on the other hand, commonly focus on word-level relations between words and their contexts. The main idea of word embeddings is that individual words are no longer treated as independent symbols, but instead they reflect similarities and correlations between words. Particularly, word embedding methods represent words as continuous vectors in a low-dimensional Euclidian space [9], [10], [11], [12], [13]. The learned embedding of a word encodes its semantic and syntactic relations with its contextual words, by utilizing local word collocation patterns. As such, the specific functions of words can also be distinguished by their embeddings, thus they can overcome the shortcomings of topic models.

As topic models and embedding methods emphasize two different types of text patterns from the text, each of them only reflects partial but complementary aspects of the whole semantics of text content. Thus, integrating topics with semantic embeddings has recently become a feasible approach to build up comprehensive text representation.

Existing methods use a dependent combination strategy that modifies structures, inputs, or assumptions of original topic models and embeddings. In order to adapt discrete topics to the continuous embedding representation, the discrete distribution assumption of topic models has been replaced with continuous distributions formed by embedding vectors [14], [15], [16]. However, this disables the topics reflecting global correlation of words and only words with big geometrical similarities between their embedding vectors will be grouped into the same topics, resulting in worse topic quality. Moreover, some models actually produce corpus-level topic embeddings or document embeddings rather than assign an embedding to each topic of a document [16], [17], [18], [19], [20]. With these models, we can not tell the meaning of each topic in a document, because they actually get no embedding vector, especially in tasks that rely on specific aspects of a document such assumption leading to worse veracity of topics. For instance, in sentiment analysis task, only small parts of topics highly related to positive or negative expressions are key text elements. Besides, some frameworks directly use the results of one model as the inputs of the other one [21], [22]. Such strategy, however, fails to explain the differences between topics and embeddings, and lacks theoretical foundations.

Specifically in this paper, we investigate the consistency of topics and word embeddings in expressing the same content of documents. We intend to make few modifications

• *Suge Wang is the corresponding author. E-mail: wsg@sxu.edu.cn*

Manuscript received xxxx, xxxx; revised xxxx, xxxx.

to the original model assumptions, thus retaining their own unique structural and functional advantages as much as possible. Additionally, we propose a novel integration framework to combine topic modeling with semantic embedding, for the proposed framework is an effective way to merge information from heterogeneous sources with different conceptual and contextual representations, that can thus make topics and embedding vectors share their high level semantic information sufficiently. The hybrid model is named as *Embedding Enhanced Topic Model*, which could assign an embedding vector to each topic and produce a topic embedding matrix for each document, rather than the corpus-level topic embeddings or document embeddings. Thus, EETM is a real topic-level representation model.

The major contribution of this work is that we find a novel way to map the topic-word's structure information of topic model into the corresponding embedding structure to generate topic-level semantic embeddings for more accurate semantic representation of topics, which is intrinsically different from researches that utilize topic information to produce high-quality word embedding or document embedding. EETM is able to describe the semantic difference of the same topic expressed in different documents, which thus can bring benefits tasks that rely on specific details or aspects of documents.

2 Related Work

In this section, we will briefly review topic models, word embedding methods, and their combinations.

Topic Models. Topic models employ hierarchical topic-word structures to explain the generative process of topics and words in text content. Particularly, hierarchical structures in topic models are usually regulated with probability distributions. Each document is assigned to the topics with different weights/probabilities, which specifies both the degree of memberships in the different topic clusters, as well as the document coordinates in the low-dimension topic space. LDA [8], known as a typical topic model, regards the mixtures of related words that are frequently used in similar text content as topics. To model the special meanings and usages of topics, categorization labels, language labels, and other semantic information are introduced into topic model models [23], [24], [25], [26], [27], [28], where words in topics are also conditioned on these labels, thus leading to better distinction of topics.

Embedding Methods. Embedding methods have been successfully applied in language models and many NLP tasks [29], [30], [31], [32]. Word embeddings are very useful because they not only can encode various text structures of syntactic and semantic information into continuous vectors, but also enable similar words locate closely in Euclidian space. Early word embedding models are time consuming due to high computational complexity, until two efficient models, namely Skip-Gram and continuous bag-of-words model (CBOW) [12], have been proposed. Semantic information at multiple levels, such as sentiment labels [33], paragraphs or sentences information [34], [35], [36], discourse structures [37], and rating information [38], can be introduced into word embeddings by forming necessary supervised structures.

Combinations of Topics and Embeddings. Recently, a couple of methods have been presented to combine topic models and word embeddings. For example, Nguyen et al. [15] proposed Latent Feature Topic Modeling (LFTM), which extends LDA to a mixture of conventional multinomial distribution and an embedding link function. However, it does not explain the differences between topic and word embeddings. Then, researchers proposed global topic embedding vectors. For instance, topical Word Embedding (TWE) [17] and Latent Topical Skip-Gram (LTSG) [18] average the embeddings of words in the same topic to get the corpus-level embedding vector of the topic. Das et al. [19] presented Gaussian LDA, which assumes that words in a topic are random samples from a multivariate Gaussian distribution with the topic embedding as the mean. In these methods, topics are modeled as global vectors for given whole data set, which ignores the real situation that topics in different documents of the data set may focus on different specific aspects and word semantics.

Some models produce overall document embedding vectors by using topic-word structures. Neural Topic Model (NTM) [14] explains the standard topic model from the perspective of a neural network, where the document is parameterized with a vector, but the vector is not directly associated to latent topics. Based on the PSDVec [9], generative topic embedding model [16] takes a document description vector as topic embedding to form link function. The efficient Correlated Topic Modeling [39] uses both global topic embeddings and document embeddings to regulate the topic weights.

Topic-based Skip-gram [21] is a semantic word embeddings architecture, which first puts the whole text corpus into topic models to capture semantic relationship between words and subsequently takes it as the input for word representation learning stage. Bidirectional Hierarchical Skip-Gram [22] is a combination of the skip-gram and topic representations, which produces overall topic embeddings based on results of the skip-gram.

3 Embedding Enhanced Topic Model

Embedding Enhanced Topic Model (EETM) is hybrid model consisted of LDA and semantic embeddings, which is implemented by applying an integration framework.

The main idea of our integration framework origins from a common cognition process in learning that one can enrich his information and knowledge by exchanging his views and opinions with others. The core of topic models and semantic embeddings is to express the content of text based on document-level word collocation and local context collocation patterns. Both of them reflect the internal semantic structure of text content, but represent this structure with topic-word distribution and linear relations of embedding vectors respectively, resulting in two perspectives of text semantics. The integration of these two methods essentially is to discover and maximize their structure consistency in expressing the same text content, with an objective to improve the text representation.

3.1 Models to be Integrated

Standard LDA [8] is the topic model integrated in EETM. It defines that text data set consists of documents, denoted as

$\boldsymbol{D} = \{d_1, d_2, ..., d_M\}$. The length of document $d_i \in \boldsymbol{D}$ is represented as N_{d_i}, and the j-th word in document d_i is notated as w_{ij}, where $j \in [1, N_{d_i}]$. The vocabulary of $\boldsymbol{D}$ is denoted as $\boldsymbol{U} = \{u_1, u_2, ..., u_W\}$. Given a document d_i, each word $w_{ij} \in d_i$ is assigned to a topic indexed by $z_{ij} \in \{1, 2, ..., K\}$, where K is the number of topics. Each topic assignment z_{ij} has a document-specific prior probability, denoted as $\theta_{ik} = P(k|d_i)$. The vector $\theta_i = (\theta_{i1}, \theta_{i2}, ..., \theta_{iK})$ is referred to as the topic distribution for document d_i. Thus, the likelihood of standard LDA given a document set $\boldsymbol{D}$ is commonly formulated as

$$\begin{aligned} &P(\boldsymbol{D}|\alpha, \beta) = p(\phi|\beta) \\ &\times \int \prod_{i=1}^{M} p(\theta_i|\alpha) \prod_{j=1}^{N_{d_i}} \sum_{z} p(z_{ij}|\theta_i) p(w_{ij}|z_{ij}, \phi) d\theta_i, \end{aligned} \tag{1}$$

where α and β is the Dirichlet prior parameters. The conditional probability $p(w_{ij}|z_{ij} = k, \phi)$ estimates the habitual usage rate of word w_{ij} for expressing topic k in document d_i, which can thus be used as the correlation measurement of word w_{ij} and topic k.

Traditional LDA groups words into topics according to the collocation patterns of words, ignoring the context backgrounds with other semantic structures, such as phrases, syntax rules, and sentential forms, which makes the topics unable to discriminate words with various linguistic functions. Thus, we introduce context information of word embeddings into topic model and increase the discriminability of topic representation.

We assume each word $u_i \in \boldsymbol{U}$ gets a pre-trained N-dimensional embedding vector $\boldsymbol{v}_{u_i}$, and all word embedding vectors form a word embedding matrix, denoted as $\boldsymbol{V} = \{\boldsymbol{v}_{u_1}, \boldsymbol{v}_{u_2}, ..., \boldsymbol{v}_{u_W}\}$. During the iterative process of our method, word embeddings remain fixed. Each topic in a document d_i has a topic embedding vector, denoted as $\boldsymbol{t}_{ik}$. Thus each document has K topic embeddings, denoted as a topic embedding matrix $\boldsymbol{T}_i = (\boldsymbol{t}_{i1}, \boldsymbol{t}_{i2}, ..., \boldsymbol{t}_{iK})$. Topic embeddings reside in the same N-dimensional space as word embeddings.

Given the j-th word w_{ij} and topic embedding matrix $\boldsymbol{T}_i$ in document d_i, the spatial relationship between word embedding $\boldsymbol{v}_{w_{ij}}$ and topic embedding $\boldsymbol{t}_{ik}$ can be defined by means of *softmax function*, which is formulated as

$$p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}}) = Softmax(\boldsymbol{t}_{ik}, \boldsymbol{v}_{w_{ij}}) = \frac{\exp(\boldsymbol{t}_{ik}^T \boldsymbol{v}_{w_{ij}})}{\sum_{l=1}^{K} \exp(\boldsymbol{t}_{il}^T \boldsymbol{v}_{w_{ij}})}. \tag{2}$$

Actually, Eq.2 calculates the intensity that word w_{ij} belongs to topic k based the semantic relevance of context information represented in the spatial space. The likelihood of topic embeddings needs a well designed model to formulate, which is not discussed in this work. For the sake of simplification, we take full connections between each word and topic in a document, and take product of Eq.2 for single topic-word pair as the likelihood. Thus, given word embeddings, the likelihood of topic embeddings is

$$P(\boldsymbol{T}|\boldsymbol{V}) = \prod_{i=1}^{M} \prod_{j=1}^{N_{d_i}} \prod_{k=1}^{K} p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}}). \tag{3}$$

3.2 Integration Framework

The conditional probabilities $p(w_{ij}|z_{ij} = k, \phi)$ and $p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}})$ provide two ways to describe semantic relations between words and topics. The information integration framework effectively connects these two types of topic-word relation measurements from topic models and embeddings by making them share consistent internal semantic structure of the text content. Note the semantic structure information acquired by topic models and embedding methods is not exactly the same. Hence they only share part of their semantic structure information, and we call it *mutual semantic information*. When a model takes in structure information from the other model, the amount of their mutual semantic information will naturally be increased. Thus, we can build up a framework to integrate topics with embedding methods by maximizing their mutual semantic information measured on their topic-word relation structure.

We first define a measurement to indicate the mutual semantic information of two text representations. The union of every topic-word pair in all documents is taken as a *semantic set* $\boldsymbol{S}$, which is formulated as

$$\boldsymbol{S} = \bigcup_{i=1}^{M} \{(w_{ij}, t_{ik}) | w_{ij} \in d_i, t_{ik} \in d_i\}. \tag{4}$$

For the sake of expressing our strategy clearly, we denote a simplified form of semantic set, which is $\boldsymbol{S} = \{s_1, s_2, ..., s_{\mathcal{N}}\}$, thus s_n represents a certain topic-word pair (w_{ij}, t_{ik}), and $\mathcal{N} = |\boldsymbol{S}|$. For a certain element $s_n \in \boldsymbol{S}$ $(1 \leq n \leq \mathcal{N})$, it gets two types of semantic measurements, notated as $\tau(s_n) = p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}})$ and $\kappa(s_n) = p(w_{ij}|z_{ij} = k, \phi)$, where $\tau(s_n) \in (0, 1)$, and $\kappa(s_n) \in (0, 1)$. Based on the idea of the *Standard Cross Entropy Loss* [40], we formulate the measurement of the mutual semantic information $G(\tau(s_n), \kappa(s_n))$ as the exponential form of binomial cross entropy loss, that is shown as follows.

Definition 1. Mutual semantic information measurement.

$$G(\tau(s_n), \kappa(s_n)) = \tau(s_n)^{\kappa(s_n)} (1 - \tau(s_n))^{1 - \kappa(s_n)} \tag{5}$$

In Definition 1, $G(\tau(s_n), \kappa(s_n))$ varies in the interval $(0, 1)$. The maximum condition of $G(\tau(s_n), \kappa(s_n))$ with regards to $\tau(s_n)$ and $\kappa(s_n)$, is essential property of the mutual semantic information measurement, where $G(\tau(s_n), \kappa(s_n))$ reaches its maximum value with following condition:

Property 1. Maximum condition of $G(\tau(s_n), \kappa(s_n))$ w.r.t. $\tau(s_n)$.

$$\tau(s_n) = \kappa(s_n) \tag{6}$$

Proof. The gradient of $\log G(\tau(s_n), \kappa(s_n))$ w.r.t. $\tau(s_n)$ is

$$\frac{\partial \log G(\tau(s_n), \kappa(s_n))}{\partial \tau(s_n)} = \frac{\kappa(s_n) - \tau(s_n)}{\tau(s_n)(1 - \tau(s_n))}. \tag{7}$$

Setting the above Eq.7 to 0 yields the optimal solution at

$$\tau(s_n) = \kappa(s_n). \tag{8}$$

■

Property 1 shows that $\tau(s_n)$ tends to share the same semantic intensity as $\kappa(s_n)$ on s_n, and the explanation of s_n learnt by measurement $\tau(.)$ tends to be consistent with $\kappa(.)$. Thus, $G(\tau(s_n), \kappa(s_n))$ could transmit semantic of $\kappa(s_n)$

into $\tau(s_n)$. On the other hand, the maximum condition of $G(\tau(s_n), \kappa(s_n))$ w.r.t. $\kappa(s_n)$, is defined as follows.

Property 2. Maximum condition of $G(\tau(s_n), \kappa(s_i))$ w.r.t. $\kappa(s_n)$.

$$\frac{\partial \log G(\tau(s_n), \kappa(s_n))}{\partial \kappa(s_n)} = \log \frac{\tau(s_n)}{1-\tau(s_n)} \begin{cases} > 0, & \text{if } \tau(s_n) > 0.5 \\ = 0, & \text{if } \tau(s_n) = 0.5 \\ < 0, & \text{if } \tau(s_n) < 0.5 \end{cases} \quad (9)$$

Property 2 shows that $G(\tau(s_n), \kappa(s_n))$ w.r.t. $\kappa(s_n)$ increases if $\tau(s_n) > 0.5$; it decreases if $\tau(s_n) < 0.5$. In other words, in the optimization process, the discriminability of $\kappa(s_n)$ could be enhanced, depending on the odds values of $\tau(s_n)$. Practically, property 2 implies the function improves the discriminability of $\kappa(.)$ by utilizing the odds of $\tau(.)$.

For the whole semantic set $\boldsymbol{S}$, we take the product of $G(\tau(s_n), \kappa(s_n))$ on each item $s_n \in \boldsymbol{S}$ as the overall *integration likelihood* w.r.t. $\tau(\boldsymbol{S})$ and $\kappa(\boldsymbol{S})$, which is formulated as follows.

$$\mathcal{G}(\tau(\boldsymbol{S}), \kappa(\boldsymbol{S})) = \prod_{n=1}^{\mathcal{N}} \tau(s_n)^{\kappa(s_n)} (1-\tau(s_n))^{1-\kappa(s_n)} \quad (10)$$

The semantic measurements $\tau(s_n)$ and $\kappa(s_n)$ are global or local semantic measurements with a certain granularity. The semantic set $\boldsymbol{S}$ connects these two measurements and provides a joint explanation of the data. During learning process, the maximization of integration likelihood has to be coordinated with the optimization of topic model and semantic embedding vectors. For a data set $\boldsymbol{D}$ and a semantic set $\boldsymbol{S}$, the likelihood of the whole integration framework is formulated as follows.

$$\mathcal{L}(\boldsymbol{D}, \boldsymbol{V}) = P(\boldsymbol{D}|\alpha, \beta) P(\boldsymbol{T}|\boldsymbol{V}) \mathcal{G}(\tau(\boldsymbol{S}), \kappa(\boldsymbol{S})), \quad (11)$$

where $P(\boldsymbol{D}|\alpha, \beta)$ and $P(\boldsymbol{T}|\boldsymbol{V})$ are likelihoods of topic model and semantic embeddings. Take logarithm of both sides in Eq.11, we obtain the logarithmic likelihood of whole framework.

$$\begin{aligned} &\log \mathcal{L}(\boldsymbol{D}, \boldsymbol{V}) \\ &= \log P(\boldsymbol{D}|\alpha, \beta) + \log P(\boldsymbol{T}|\boldsymbol{V}) + \log \mathcal{G}(\tau(\boldsymbol{S}), \kappa(\boldsymbol{S})) \end{aligned} \quad (12)$$

Eq.12 can be interpreted as the logarithmic likelihood of data set $\boldsymbol{D}$ collaboratively described by topics and semantic embeddings. The consistency of these two text representations is measured on semantic set $\boldsymbol{S}$. By maximizing Eq.12, semantic information can be transmitted across representations through the mutual semantic information function $G(.,.)$. The topics and embeddings could acquire additional information from each other and produce better representation of data.

3.3 Generative Process

The key idea of a topic model is to explain the generative process of each word in documents. When word and topic embeddings are included in the topic model, the generative process also involves the probability structure of word embeddings. In EETM, the generative process of words in documents can be regarded as a hybrid of LDA and the semantic embeddings together with integration framework.

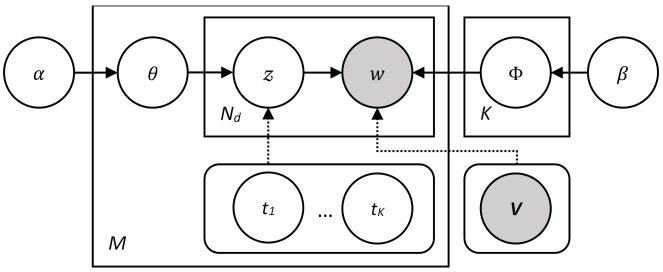


Fig. 1. Plate notation of EETM. EETM involves the generative process of standard LDA and additional semantic embeddings. Topics and words are assigned with corresponding topic embedding and word embedding vectors (dot line). Thus, each document gets a topic embedding matrix including K topic embedding vectors. Word embeddings are pre-trained and fixed during the learning process of EETM. A certain word w_{ij} is generated according to $p(w_{ij}|z_{ij}, \phi, \boldsymbol{T}_i, \boldsymbol{V})$.

As the word embedding $\boldsymbol{v}_{s_i}$ is produced by a certain word embedding method, we ignore its producing process here and focus on the generative process of words along with topic embeddings in documents. Based on Eq.11, the generative process of document d_i is as follows:

1. Draw $N_{d_i} \sim Poisson(\xi)$;
2. Draw $\theta_i \sim Dir(\alpha)$;
3. For the k-th topic, draw a topic embedding uniformly from a hyper-ball of radius μ, i.e. $p(\boldsymbol{t}_{ik}) = \frac{3}{4\pi\mu^3}$ if $||\boldsymbol{t}_{ik}||_2 \leq \mu$, 0 otherwise;
4. For each word w_{ij} in document d_i:
   (a) Draw a topic $z_{ij} \sim Multinomial(\theta_i)$;
   (b) Draw word w_{ij} from $\boldsymbol{U}$ according to $p(w_{ij}|z_{ij}, \phi, \boldsymbol{T}_i, \boldsymbol{V})$.

The conditional probability $p(w_{ij}|z_{ij}, \phi, \boldsymbol{T}_i, \boldsymbol{V})$ represents the distribution of w_{ij}. It can be determined by our proposed integration framework, which will be described in following subsections. Above generative process is presented in plate notation in Figure 1.

3.4 Explanation

As shown in Eq.12, the likelihood of EETM includes likelihood of LDA and topic embeddings, while integration likelihood connects them. The original motivation of our strategy is to maximize the mutual semantic information of two models and make them share most of their knowledge on the data set. Thus, the mutual semantic information has to be optimized with the two models simultaneously. Furthermore, the mutual semantic information itself can also be interpreted in a conditional probabilistic way, which could explain the relationship between the two models.

The integration framework includes two ways of topic-word relation modeling: the topic distribution of LDA and the spatial intensity of embeddings. Eq.3 formulates the probability of topic with condition on word embeddings. When focusing on the topic intensity measured by topic and word embeddings, $p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}})$ and $1 - p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}})$ form up a binomial distribution of the topic assignment to a word. For each topic-word pair in $\boldsymbol{S}$, the mutual semantic information function (Eq.5) can be regarded as a prior of the binomial distribution, and the prior information comes from LDA. Specifically, the prior information is topic-word distribution $p(w_{ij}|z_{ij} = k, \phi)$, which is a part of document

modeling in LDA, and the likelihood $P(\boldsymbol{D}|\alpha,\beta)$ involves $p(w_{ij}|z_{ij}=k,\phi)$ as a component. Hence we can hold the prior information from $P(\boldsymbol{D}|\alpha,\beta)$, and the integration likelihood $\mathcal{G}(\kappa(\boldsymbol{S}),\tau(\boldsymbol{S}))$ can be interpreted as a conditional probability w.r.t. $\boldsymbol{T}$.

$$\mathcal{G}(\kappa(\boldsymbol{S}),\tau(\boldsymbol{S}))=P(\boldsymbol{T}|\boldsymbol{D}) \tag{13}$$

Eq.13 contains prior terms for every $p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}})$ in Eq.3. Thus, $P(\boldsymbol{T}|\boldsymbol{D})$ can be regarded as prior for $P(\boldsymbol{T}|\boldsymbol{V})$, and we obtain

$$P(\boldsymbol{T}|\boldsymbol{D},\boldsymbol{V})=P(\boldsymbol{T}|\boldsymbol{D})P(\boldsymbol{T}|\boldsymbol{V}), \tag{14}$$

where $\boldsymbol{D}$ and $\boldsymbol{V}$ are independent variables. Plugging Eq.14 into Eq.11, the likelihood of EETM can also be interpreted as

$$\begin{aligned}\mathcal{L}(\boldsymbol{D},\boldsymbol{V})&=P(\boldsymbol{D}|\alpha,\beta)P(\boldsymbol{T}|\boldsymbol{V})\mathcal{G}(\kappa(\boldsymbol{S}),\tau(\boldsymbol{S}))\\&=P(\boldsymbol{D}|\alpha,\beta)P(\boldsymbol{T}|\boldsymbol{D},\boldsymbol{V})\\&=P(\boldsymbol{D},\boldsymbol{T}|\alpha,\beta,\boldsymbol{V}),\end{aligned} \tag{15}$$

where topic embeddings $\boldsymbol{T}$ and topic-word distribution of data set $\boldsymbol{D}$ are learning objectives of EETM.

As for each word w_{ij} in document d_i, there is a corresponding topic assignment z_{ij}. When focusing on the topic-word distribution, it depends on the global topic-word distribution and topic intensity of embeddings. For each topic-word pair in $\boldsymbol{S}$, it has a corresponding topic intensity measurement from the mutual semantic information function. Thus, given a certain topic assignment $z_{ij}=k$, the word distribution $p(w_{ij}|z_{ij},\phi,\boldsymbol{T}_i,\boldsymbol{V})$ can be formulated as

$$\begin{aligned}&p(w_{ij}|z_{ij}=k,\phi,\boldsymbol{T}_i,\boldsymbol{V})=p(w_{ij}|z_{ij}=k,\phi)\\&\times p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}})^{p(w_{ij}|z_{ij}=k,\phi)}\\&\times(1-p(\boldsymbol{t}_{ik}|\boldsymbol{v}_{w_{ij}}))^{1-p(w_{ij}|z_{ij}=k,\phi)}.\end{aligned} \tag{16}$$

Eq.16 can be interpreted as the generative distribution of word w_{ij} in the generative process of EETM.

4 Learning Algorithm

EETM is a hybrid model consisting of LDA and semantic embeddings, and it contains two kinds of parameters: distribution parameters and embedding vectors. Our proposed learning process for EETM includes two parts: maximum optimization of the corpus probability, and the approximation of the topic embeddings. We adapt the *Generalized Expectation Maximization* (GEM) algorithm to optimize the likelihood of EETM (Eq.12).

Specifically, during the optimization of topic-word distributions, the topic embeddings are assumed to be known and constant. During the approximation of topic embeddings, however, the topic-word distributions are utilized as conditions. These above two parts are repeated until convergence. Finally, the learning algorithm outputs topic-word distributions and topic embeddings for each document.

4.1 Variational Inference

EETM contains latent variables θ and z, which can not be estimated directly. In this paper, we utilize *Variational Inference* [41] to seek the optimal $\mathcal{L}(\boldsymbol{D},\boldsymbol{T})$. As stated in Eq.15, we treat $P(\boldsymbol{D},\boldsymbol{T}|\alpha,\beta,\boldsymbol{V})$ equally as $\mathcal{L}(\boldsymbol{D},\boldsymbol{T})$ in the following equations.

For an arbitrary variational distribution $q(\theta,z)$, the following equalities hold

$$\begin{aligned}&E_q[\log\frac{P(\boldsymbol{D},\boldsymbol{T},\theta,z|\alpha,\beta,\boldsymbol{V})}{q(\theta,z)}]\\=&E_q[\log P(\boldsymbol{D},\boldsymbol{T},\theta,z|\alpha,\beta,\boldsymbol{V})]+H[q]\\=&\log P(\boldsymbol{D},\boldsymbol{T}|\alpha,\beta,\boldsymbol{V})-KL(q||p),\end{aligned} \tag{17}$$

where $p=p(\theta,z|\alpha,\beta,\boldsymbol{V})$, $H[q]$ is the entropy of q, and $KL(q||p)$ is the *Kullback-Leibler divergence* of p and q. This implies

$$\begin{aligned}&KL(q||p)\\=&\log P(\boldsymbol{D},\boldsymbol{T}|\alpha,\beta,\boldsymbol{V})\\&-(E_q[\log P(\boldsymbol{D},\boldsymbol{T},\theta,z|\alpha,\beta,\boldsymbol{V})]+H[q])\\=&\log P(\boldsymbol{D},\boldsymbol{T}|\alpha,\beta,\boldsymbol{V})-\mathcal{L}(q,\boldsymbol{T}).\end{aligned} \tag{18}$$

In Eq.18, $E_q[\log P(\boldsymbol{D},\boldsymbol{T},\theta,z|\alpha,\beta,\boldsymbol{V})]+H[q]$ is usually referred to as the *variational free energy* $\mathcal{L}(q,\boldsymbol{T})$, which is a lower bound of $\log P(\boldsymbol{D},\boldsymbol{T}|\alpha,\beta,\boldsymbol{V})$. It is intractable to directly maximizing $\log P(\boldsymbol{D},\boldsymbol{T}|\alpha,\beta,\boldsymbol{V})$ due to the hidden variables θ and z. So we maximize its lower bound $\mathcal{L}(q,\boldsymbol{T})$ instead. We adopt a mean-field approximation of the true posterior as the variational distribution, and use a variational algorithm to find q^* maximizing $\mathcal{L}(q,\boldsymbol{T})$.

The mean-field assumption of variational distribution is formulated as

$$\begin{aligned}&q(\theta,z;\gamma,\varphi)=q(\theta;\gamma)q(z;\varphi)\\&=\prod_{i=1}^{M}\{Dir(\theta_i;\gamma_i)\prod_{j=1}^{N_{d_i}}Multinomial(z_{ij};\varphi_{ij})\}.\end{aligned} \tag{19}$$

Bringing Eq.19 into the variational free energy, we can obtain the objective function with respect to q and $\boldsymbol{T}$.

$$\begin{aligned}&\mathcal{L}(q,\boldsymbol{T})\\=&E_q[\log P(\boldsymbol{D},\boldsymbol{T},\theta,z|\alpha,\beta,\boldsymbol{V})]+H[q]\\=&E_q[\log p(\phi|\beta)]+\sum_{i=1}^{M}\{E_q[\log p(\theta_i|\alpha)]\\&+\sum_{j=1}^{N_{d_i}}E_q\left[\log\sum_z p(z_{ij}|\theta_i)p(w_{ij}|z_{ij},\phi)\right]\\&+\sum_{j=1}^{N_{d_i}}\sum_{k=1}^{K}\{E_q[\log G(\boldsymbol{t}_{ik},\boldsymbol{v}_{w_{ij}})]\\&+E_q[\log\tau(\boldsymbol{t}_{ik},\boldsymbol{v}_{w_{ij}})]\}\}+H(q)\end{aligned} \tag{20}$$

Introducing the *Jensen Inequation* to Eq.20, we obtain the lower bound of $\mathcal{L}(q,\boldsymbol{T})$.

$$\begin{aligned}&\mathcal{L}(q,\boldsymbol{T})\\\geq&E_q[\log p(\phi|\beta)]+\sum_{i=1}^{M}\{E_q[\log p(\theta_i|\alpha)]\\&+\sum_{j=1}^{N_{d_i}}\sum_z\{E_q[\log p(z_{ij}|\theta_i)]\\&+E_q[\log p(w_{ij}|z_{ij},\phi)]\}\\&+\sum_{j=1}^{N_{d_i}}\sum_{k=1}^{K}\{E_q[\log\mathcal{G}(\boldsymbol{t}_{ik},\boldsymbol{v}_{w_{ij}})]\\&+E_q[\log\tau(\boldsymbol{t}_{ik},\boldsymbol{v}_{w_{ij}})]\}\}+H(q)\end{aligned} \tag{21}$$

The further details on derivation of Eq.21 are introduced in Appendix A, where Eq.31 formulates the expansion of each item in Eq.21.

We proceed to optimize Eq.21 with a Generalized Expectation Maximization (GEM) algorithm. The update equations of the GEM algorithm will be introduced in the following subsection.

4.2 Update Equations

The GEM algorithm includes E-Step and M-Step. Particularly, in E-Step, for the l-th iteration, both $\boldsymbol{T} = \boldsymbol{T}^{(l-1)}$ and $\phi = \phi^{(l-1)}$ are treated as constants. By maximizing the objective function $\mathcal{L}(q, \boldsymbol{T})$ w.r.t. γ_i and φ_{ij} respectively, as described in Appendix B, we obtain the optimal solutions:

$$\varphi_{ijk} \propto \frac{\tau(\boldsymbol{t}_{ik}, \boldsymbol{v}_{w_{ij}})}{1 - \tau(\boldsymbol{t}_{ik}, \boldsymbol{v}_{w_{ij}})} \phi_{kw_{ij}} \exp\{\psi(\gamma_{ik})\}, \tag{22}$$

$$\gamma_{ik} = \alpha + \sum_{j=1}^{N_{d_i}} \varphi_{ijk}, \tag{23}$$

where $\psi(.)$ is the *digamma function*.

In M-Step, on the other hand, for the l-th iteration, both $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(l-1)}$ and $\boldsymbol{\varphi} = \boldsymbol{\varphi}^{(l-1)}$ are treated as constants. By maximizing the objective function $\mathcal{L}(q, \boldsymbol{T})$ w.r.t. ϕ, as introduced in Appendix C, we obtain the optimal solution:

$$\phi_{ku_n} \propto \beta + \sum_{i=1}^{M} \sum_{j=1}^{N_{d_i}} w_{ij}^{u_n} \varphi_{ijk}, \tag{24}$$

where $w_{ij}^{u_n} = 1$ only if $w_{ij} = u_n$; 0 otherwise.

To update $\boldsymbol{T}$, we first take the derivative of Eq.21 w.r.t. $\boldsymbol{t}_{ik}$, and then employ the *Gradient Descent Method* to optimize the learning objective. As elaborated in Appendix C, the gradient of $\mathcal{L}(q, \boldsymbol{T})$ w.r.t. $\boldsymbol{t}_{ik}$ is obtained as

$$\frac{\partial \mathcal{L}(q, \boldsymbol{T})}{\partial \boldsymbol{t}_{ik}} = \sum_{j=1}^{N_{d_i}} (1 + \varphi_{ijk} - 2\tau(\boldsymbol{t}_{ik}, \boldsymbol{v}_{w_{ij}})) \boldsymbol{v}_{w_{ij}}. \tag{25}$$

Then the iterative formulation of $\boldsymbol{t}_{ik}$ is defined as

$$\boldsymbol{t}_{ik}^{(l+1)} = \boldsymbol{t}_{ik}^{(l)} + \delta \frac{\partial \mathcal{L}(q, \boldsymbol{T})}{\partial \boldsymbol{t}_{ik}}, \tag{26}$$

where δ is the learning rate. To satisfy the constraint that $||\boldsymbol{t}_{ik}||_2 \leq \mu$, when $||\boldsymbol{t}_{ik}||_2 > \mu$, we normalize $\boldsymbol{t}_{ik}$ by $\mu / ||\boldsymbol{t}_{ik}||_2$.

4.3 GEM Algorithm

The *Generalized Expectation Maximization* (GEM) algorithm utilized in the optimization of EETM is illustrated as Algorithm 1. The GEM algorithm is a combination of *Variational EM algorithm* and *Gradient Descent Method*. In E-Step of GEM algorithm, the posterior distribution of latent topics in each document is estimated, during which other parameters and embedding vectors are fixed. Then in the M-Step, word distribution parameters and topic embeddings are optimized.

We put some additional termination conditions to the iterative steps and the overall procedure for GEM algorithm. The maximum number of iterative steps is set to $\mathcal{T}_{itv}$, which controls the convergence process (Line 11 in E-Step) and gradient descent process (Line 3 in M-Step). Moreover, the maximum iterative number of overall procedure is set to $\mathcal{T}_{all}$. Thus, the iterative number of E-Step is no more than $(M \times \mathcal{T}_{all} \times \mathcal{T}_{itv} \times \bar{N}_d \times K)$, and the iterative number of M-Step is less than or equal to $(M \times \mathcal{T}_{all} \times \mathcal{T}_{itv} \times K)$, where $\bar{N}_d$ is the average length of documents. As $\mathcal{T}_{all}$, $\mathcal{T}_{itv}$, and K are fixed parameters, the running complexity of GEM algorithm is $O(M \times \bar{N}_d)$.

For E-Step and Gradient Descent Method (Line 3 in M-Step), documents are independent of each other, which makes it possible for applying parallel computing against documents. For topic-word distribution updating steps (Line 1 and Line 2 in M-Step), topics are independent of each other as well. Therefore, by utilizing parallel programming and acceleration techniques in numerical calculation, the GEM algorithm could work efficiently.

Algorithm 1 GEM algorithm for EETM.

E-Step: For each document d_i, compute the variational parameters $\{\gamma_i^*, \varphi_i^*\}$.

1: Initialize $\varphi_{ijk}^{(0)} := 1/K$, $j \in [1, N_{d_i}]$, $k \in [1, K]$;
2: Initialize $\gamma_{ik}^{(0)} := \alpha + N_{d_i}/K$, $k \in [1, K]$;
3: **repeat**
4: **for** $j = 1$ to N_{d_i} **do**
5: **for** $k = 1$ to K **do**
6: $\varphi_{ijk}^{(l+1)} := \frac{\tau(\boldsymbol{t}_{ik}, \boldsymbol{v}_{w_{ij}})}{1-\tau(\boldsymbol{t}_{ik}, \boldsymbol{v}_{w_{ij}})} \phi_{kw_{ij}} \exp\{\psi(\gamma_{ik}^{(l)})\}$;
7: **end for**
8: Normalize $\varphi_{ij}^{(l+1)}$ to sum to 1;
9: **end for**
10: $\gamma_{ik}^{(l+1)} := \alpha + \sum_{j=1}^{N_{d_i}} \varphi_{ijk}^{(l+1)}$;
11: **until** convergence.

M-Step: Update model parameters ϕ, and topic embedding matrix $\mathbf{T_i}$.

1: Update $\phi_{ku_n} := \beta + \sum_{i=1}^{M} \sum_{j=1}^{N_{d_i}} w_{ij}^{u_n} \varphi_{ijk}$;
2: Normalize ϕ_k to sum to 1;
3: Update each topic embedding $\boldsymbol{t}_{ik}$ in document d_i with *Gradient Descent Method*.

5 Experimental Results

In our proposed EETM, each topic in a given document d_i receives an embedding vector $\boldsymbol{t}_{ik}$ and a conditional probability $\theta_{ik} = P(k|d_i)$, which can be used to build up the text representation in several text analysis tasks.

In our experiments, we have studied the performance of EETM by setting up three different text representations, i.e. θ_i, $\boldsymbol{t}_i$, and the combination of θ_i and $\boldsymbol{t}_i$. Particularly, we first select optimal parameters for EETM. Then, to demonstrate the topic coherence of EETM, we present the top words in topics by comparing with conventional topic models. Finally, we investigate the document representation quality of EETM by testing it on several common text analysis tasks.

5.1 Experimental Setup

Data Sets We employed three corpora for evaluation: the 20Newsgroups[1] (20NG) and the Reuters-21578 corpus[2]

1. http://qwone.com/~jason/20Newsgroups/
2. http://www.nltk.org/book/ch02.html

(Reuters), and Hotel Reviews[3] (Hotel). For Reuters, we removed the documents appearing in two or more categories, and selected the largest 10 categories that contain to 8,025 documents, following [16]. Hotel Reviews corpus contains 4,000 English hotel reviews with sentiment labels, i.e. 2,000 positive documents, and 2,000 negative documents. Hotel Reviews is a short text corpus whose averaged length is 42 words. Note the same preprocessing steps are applied to all data sets, e.g. convert all words into lower cases, remove stop words[4] and those words out of the word embedding vocabulary.

Compared methods While the above three data sets have class labels, our proposed EETM, as an *unsupervised* text representation method, does not use the class labels in our learning process. Instead, we only utilize the labels for evaluation purpose. Three text representations built up by utilizing the results of EETM are evaluated, including:

- **TR**: the topic representation with optimized γ_i^* learned by EETM;
- **TE**: the topic embeddings learned by EETM, which are produced by the concatenation of all topic embeddings in a document;
- **TR+TE**: the concatenation of the TR and TE.

Baseline Methods We compare our proposed EETM against seven unsupervised state-of-the-art text modeling methods, including three topic modeling methods, three text representation methods and one conventional method. In particular, three topic models compared in our experiments are:

- **LDA**: the LDA [8] implemented in gensim library[5];
- **GaussianLDA**: Gaussian LDA[6] [19], whose posterior topic distribution is utilized as document representation;
- **LFTM**: Latent Feature Topic Modeling[7] [15], which provides topic distribution in a document as the representation.

Three text representation methods are:

- **Doc2Vec**: Paragraph Vector [42] in gensim library[8];
- **TWE**: Topical Word Embedding[9] [17], and the averaged topic embedding is used as document representation;
- **WE**: the mean word embedding of the document.

For further comparison, concatenations of LDA and word embedding (**LDA+WE**) are included in our experiment. In addition, conventional bag-of-word (**BOW**) method is included in our baselines, where each element in a vector is a binary value, indicating whether it has the word or not.

Experimental Settings We leveraged three pre-trained word embeddings in our experiments, namely, GloVe[10], Word2Vec[11] with Skip-Gram (**Skip**), and CBOW (**CBOW**) model, respectively. The pre-trained word embeddings were learnt on a March 2017 Wikipedia snapshot[12]. They contain the most frequent 224,500 words. The dimensionalities of word embeddings and topic embeddings are set to 50 and 100, following [17].

The hyperparameters of topic model, including EETM and baselines, are set as $\alpha = 50/K$, $\beta = 200/|\boldsymbol{U}|$. The radius of topic embeddings μ is set to 7 [16]. Thresholds of iterative number are set as $\mathcal{T}_{itv} = 100$, $\mathcal{T}_{all} = 300$. During the experiments, we find the learning rate δ has little influence on the performance of EETM. Thus, the learning rate δ is set to 0.5 for faster convergence. Samples in each data set are represented by EETM, ignoring their category labels. After about 200 GEM iterations on each data set, topic distributions and topic embeddings are obtained.

5.2 Topic Coherence

To evaluate topic coherence of EETM, we employ the NPMI [43] as our metric. The number of topics is set to 20 initially, and increased by 20 until 100. For each topic, top 10 words are used to calculate NPMI value, after which the averaged NPMI value of all topics are employed for evaluation. Among baselines, only LDA could provide overall topic-word distribution, thus we compared EETM with LDA. Results of standard LDA and EETM with three pre-trained word embedding methods (whose dimensionalities are 50 and 100 respectively) are listed in Table 1, where best performance is in bold face.

Higher value of NPMI reflects that the top words within topics are more strongly related to each other, which means better quality of topics. As we can see in Table 1, EETM presents much better quality of topic coherence than LDA especially when the number of topic is set to lower value. As EETM calculates a topic semantic embedding matrix for every document, lower number of topic can effectively reduce the computation cost. In the learning process of EETM, topic embeddings are fixed during E-Step, thus the computation cost of E-Step is generally close to standard LDA. While in M-Step of GEM algorithm, topic embeddings are updated by Gradient Descent Method (GDM). Based on our experiment, we find that the iterative precision of topic embeddings in the early iterations of GEM algorithm is not crucial to the final result. When the topic distribution is close to convergence state, topic embeddings can more easily reach the convergence state. Even topic embeddings of the longest document from 20NG can reach the convergence state (precision threshold is 0.0001) within 200 GDM iterations. Thus, in our experiment, maximum GDM iterative times is limited to 200. After optimization, overall time cost of EETM is acceptable.

Another phenomenon presented in Table 1 is that three pre-trained word embedding methods generally produce similar results and lower dimensionality of word embeddings can show competitive or best topic coherence. This makes clear that EETM is *not sensitive* to the dimensionality of semantic embedding vectors, which is intrinsically different from other embedding-learning methods. Thus, any

3. http://www.liip.cn/ccir2014/pc.html
4. http://www.nltk.org/book/ch02.html
5. http://radimrehurek.com/gensim/models/ldamulticore.html
6. https://github.com/rajarshd/Gaussian_LDA
7. https://github.com/datquocnguyen/LFTM/
8. http://radimrehurek.com/gensim/models/doc2vec.html
9. https://github.com/largelymfs/topical_word_embeddings/
10. https://nlp.stanford.edu/projects/glove/
11. http://radimrehurek.com/gensim/models/word2vec.html
12. https://dumps.wikimedia.org/

TABLE 1
NPMI comparison of EETM and LDA.

Method	20NG					Reuters					Hotel				
	20	40	60	80	100	20	40	60	80	100	20	40	60	80	100
EETM(CBOW50)	**1.089**	1.124	1.477	1.485	1.649	1.143	1.362	1.614	**1.738**	**1.832**	0.450	0.724	**1.008**	**1.171**	1.323
EETM(Skip50)	0.978	1.167	1.469	1.597	1.701	1.214	1.396	1.392	1.633	1.628	0.472	**0.927**	1.003	1.061	**1.373**
EETM(Glove50)	0.728	1.076	1.226	1.453	1.632	1.109	1.175	1.244	1.488	1.559	**0.540**	0.883	0.948	1.052	1.177
EETM(CBOW100)	0.993	1.320	1.397	1.541	1.776	**1.216**	1.419	**1.693**	1.665	1.811	0.431	0.846	0.987	1.103	1.212
EETM(Skip100)	0.996	**1.355**	**1.495**	**1.726**	**1.791**	1.204	**1.446**	1.545	1.634	1.751	0.460	0.716	0.906	1.077	1.134
EETM(Glove100)	0.874	1.128	1.090	1.432	1.666	0.905	1.089	1.300	1.354	1.611	0.524	0.783	0.972	1.045	1.143
LDA	0.178	0.290	0.392	1.209	1.783	0.761	0.798	0.859	1.304	1.645	0.484	0.532	0.507	0.425	0.320

TABLE 2
Top words in most representative topics of EETM and LDA.

Data Set	EETM (CBOW)					EETM (Skip)				
20NG	edu	president	graphics	key	medical	edu	president	graphics	key	medical
	technology	clinton	edu	clipper	harvard	posting	clinton	edu	des	pitt
	institute	stephanopoulos	3d	chip	medicine	host	stephanopoulos	3d	pgp	disease
	columbia	myers	ray	encryption	health	nntp	going	ray	public	gordon
	insurance	george	bbs	keys	treatment	organization	myers	pub	bit	cancer
Reuters	oil	foreign	coffee	dollar	billion	oil	bank	coffee	west	billion
	gas	government	export	west	pct	gas	banks	export	exchange	january
	tax	exchange	brazil	exchange	dlrs	crude	loan	quotas	dollar	february
	pct	debt	quotas	rates	year	texas	loans	brazil	paris	year
	production	banks	meeting	policy	january	barrel	interest	ico	baker	rose
Hotel	very	room	location	good	not	very	room	nice	good	not
	poor	bathroom	helpful	nice	bed	staff	bed	location	friendly	too
	all	big	hotel	friendly	work	breakfast	bathroom	also	great	even
	fantastic	use	staff	well	cleaned	helpful	there	little	comfortable	poor
	center	enough	near	excellent	smoking	excellent	small	walking	location	no
Data Set	EETM (Glove)					LDA				
20NG	edu	president	graphics	key	medical	make	dean	dean	organization	answers
	posting	will	edu	number	health	image	organization	worshipped	dean	dean
	host	clinton	ray	bit	disease	dean	bible	covenant	whether	group
	nntp	stephanopoulos	3d	chip	cancer	correct	answers	quiz	com	god
	organization	press	art	bits	patients	organization	timmbake	mexico	suggesting	only
Reuters	oil	bank	coffee	exchange	billion	recovery	quick	recovery	bank	merger
	crude	banks	export	west	yen	merger	sumitomo	quick	merger	bank
	gas	loans	quotas	dollar	assets	bank	bank	sumitomo	quick	sumitomo
	barrels	interest	ico	baker	loans	financial	recovery	bank	recovery	quick
	ecuador	credit	brazil	paris	deposits	japan	sumi	small	sumitomo	recovery
Hotel	very	room	location	polite	not	people	good	reach	sized	value
	staff	bathroom	great	staff	room	location	great	easily	people	hotel
	helpful	shower	hotel	helpful	night	great	really	tourist	very	sized
	friendly	small	other	friendly	reception	sized	without	locations	location	problems
	well	water	near	well	air	good	area	heart	easily	good

word embedding of high quality can make EETM work well. In order to illustrate the effect of pre-trained word embeddings, we lists top words in most representative topics learned by EETM and LDA, shown in Table 2, where each column presents topic words within one topic.

20NG is a data set containing emails from 20 different news groups. As shown in Table 2, related words are clustered into same topics, and these topics can be clearly recognized as *technology*, *education*, *politics*, and *medicine*. Similar results can be observed in Reuters and Hotel. Particularly, Reuters covers topics of *economics*, *trade*, and *financial policy*. Thus, EETM can discover topics effectively, indicating the original functions of LDA have been retained well in EETM. Interestingly, words with specific syntactic functions were also clustered into individual topics, e.g. quantifiers and months, which are contained in the Reuters news, can be assigned into the same topic, i.e. the fifth column of Reuters, as they have similar syntactic or structural functions in sentences, indicating EETM is capable to capture additional structure information from pre-trained word embeddings.

Hotel is a review data set that contains sentiment terms and expressions. Topics learned from Hotel data include *aspects* of reviews, such as *service*, *location*, and *room*. In those topics focusing on different aspects of hotels, words frequently used to describe the same aspect of a topic are clustered together. As for sentiments of reviews, words expressing positive and negative opinions are split into individual topics. Interestingly, the fourth topic (the fourth column) is associated to positive words, while negative words are clustered in the fifth topic (the fifth column). These sentiment topics tend to cover adjectives and adverbs with high probabilities, indicating that EETM can coordinate topics and word embeddings to leverage both topic information and syntactic/structural information.

Comparing the words from three pre-trained word embeddings, we observe EETM performs similarly and consis-

tently. In other words, corresponding topics learned from three word embeddings share a large portion of their top words, with small differences in the order of words. In most cases, the first several words are the same. The results show EETM is stable and reliable no matter which word embedding methods are selected. By integrating the semantic information of word embeddings with topics, EETM benefits from both of them to select similar top words expressing a certain opinion to form a topic.

Conventional topic models typically need a large data set of documents with various content to extract the coherent topic space. However, when the given data set is not big enough or lacks of content diversity, conventional topic models will suffer. The three data sets used in our experiments are comparatively small data sets. In addition, categories of Reuters are imbalanced, and Hotel is a domain-specified data set. Nevertheless, EETM could acquire enough semantic information from pre-trained word embeddings, and it is thus able to extract effective topics even with small or domain-specified data sets. However, traditional LDA model suffers from smaller data set and produces inferior results. Specifically, we observe many words expressing different aspects are mixed in the same topics, while different topics identified express same or similar content.

5.3 Parameter Selection

Following evaluations in our experiment include classification task and clustering task. Thus, we perform 5-fold cross validation (5FCV) on each data set to find the optimal K value. To evaluate the performance of different K values, we train ℓ-1 regularized linear SVM classifier in one-vs-all fashion [13] and employ the average accuracy on the test sets in 5FCV as the evaluation metric.In order to test the overall performance of our proposed EETM method, during the parameter selection process, TR+TE has been used for evaluation. This is because TR+TE, leveraging both topic models and word embeddings, is able to perform better than TR or TE individually.

Table 3 lists the optimal parameters on three data sets with three word embedding methods, in terms of topic number K and the dimensionality of word embedding N.

TABLE 3
Optimal parameters for EETM.

Data Sets	w2v(CBOW)		w2v(Skip)		GloVe	
	K	N	K	N	K	N
20NG	80	100	100	100	80	100
Reuters	40	50	60	50	80	50
Hotel	40	50	20	50	20	50

In the following qualitative assessment, all EETM methods will use the parameters in Table 3. For fair comparison with our baseline topic modeling methods, the number of topics is set to the maximum value of EETM on three pre-trained word embedding methods in order to get lower perplexity for baseline methods, that is 100, 80 and 40 for 20NG, Reuters and Hotel, respectively. Other parameters in our baselines are set to their default values that generally lead to better results.

13. http://scikit-learn.org/stable/modules/svm.html

5.4 Evaluation on Text Analysis Tasks

We are now ready to evaluate the performance of EETM on three common text analysis tasks, including text categorization, sentiment classification, and text clustering.

Both text categorization and sentiment classification are classification tasks. In particular, 20NG and Reuters are multi-class data sets consisting of 20 and 10 classes respectively, and we use them in the text categorization task. Hotel is a review data set with binary sentiment labels, which is used in the binary sentiment classification task. All of the three data sets have also been used in the clustering task by ignoring their class labels.

5.4.1 Classification Task Evaluation

We perform *5-fold cross validation* for each data set. The standard macro-averaged precision (Prec), recall (Rec), F1 measurement (F1), and accuracy (Acc) [8], [17], [24], which are widely used for comparing the performance among different classifiers, are employed as the evaluation metrics. The performance of different methods is listed in Table 4. We include our proposed EETM results with three configurations, i.e. TR, TE, and TR+TE. Best performance in Table 4 are listed in bold. We observe that EETM achieves the best performance, comparing with other state-of-the-art methods.

TR with three pre-trained word embedding methods outperforms LDA across three data sets, indicating that the proposed information integration strategy works well on transmitting semantics of word embeddings into topics. The reason that TR outperforms LDA is it can balance the word collocation patterns and context patterns, and topic distribution can thus express context information as well.

GaussianLDA also uses word embeddings to enhance topics. However, it changes LDA's topic-word distribution assumptions. Thus, it fails to extract effective word collocation patterns, leading to its inferior performance. In comparison, our EETM can retain the original assumptions of LDA and thus inherit advantages of LDA naturally. Furthermore, GaussianLDA takes the mean of word embeddings as the corresponding topic embedding, in which Euclidean distance is used as semantic measurement of word embeddings and topic embeddings, leading to the result that similar embedding vectors will be grouped into same topics. However, high similarities of embedding vectors only indicate these words have highly similar syntax functions and context backgrounds, which does not mean they are frequently used to express the same topic. Thus, GaussianLDA is not able to discover coherent topics.

Topic embeddings of EETM are constructed by the embeddings of related words in corresponding topics, where important word embeddings get higher weights. Thus, they focus on the specific content of each topic in a document. In Table 4, TE with three pre-trained word embedding methods outperforms WE with the same word embedding methods as well as Doc2Vec, for WE and Doc2Vec mix the words of different topics into an overall embedding vector, leading to its failure to distinguish the content from different topics.

TABLE 4
Classification performance comparison on SVM.

Method	20NG				Reuters				Hotel			
	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc
BOW	0.736	0.734	0.735	0.731	0.876	**0.857**	0.866	0.936	0.899	0.932	0.916	0.914
LDA	0.754	0.777	0.765	0.756	0.859	0.830	0.844	0.928	0.866	0.887	0.877	0.875
GaussianLDA	0.398	0.424	0.411	0.413	0.448	0.266	0.334	0.721	0.806	0.510	0.625	0.694
LFTM	0.778	**0.798**	0.788	0.778	0.863	0.845	0.854	0.935	0.907	0.913	0.910	0.909
Doc2Vec	0.408	0.417	0.412	0.410	0.613	0.334	0.433	0.660	0.772	0.814	0.792	0.787
TWE	0.759	0.782	0.770	0.761	0.862	0.831	0.846	0.927	0.852	0.878	0.865	0.862
WE(CBOW)	0.606	0.627	0.616	0.611	0.817	0.551	0.658	0.846	0.902	0.903	0.903	0.902
WE(Skip)	0.615	0.639	0.627	0.623	0.810	0.573	0.671	0.852	0.894	0.896	0.895	0.895
WE(GloVe)	0.608	0.630	0.619	0.614	0.796	0.545	0.647	0.842	0.888	0.886	0.887	0.887
LDA+WE(CBOW)	0.775	0.765	0.770	0.779	0.877	0.847	0.862	0.936	0.926	0.926	0.926	0.926
LDA+WE(Skip)	0.774	0.766	0.770	0.779	0.881	0.852	0.866	**0.937**	0.926	0.925	0.925	0.925
LDA+WE(Glove)	0.775	0.763	0.769	0.776	0.882	0.850	0.866	**0.937**	0.916	0.916	0.916	0.916
TR(CBOW)	0.744	0.754	0.749	0.741	0.854	0.798	0.825	0.916	0.911	0.933	0.922	0.921
TR(Skip)	0.754	0.762	0.758	0.748	0.884	0.809	0.845	0.923	0.923	0.944	0.933	0.932
TR(GloVe)	0.764	0.772	0.768	0.757	0.872	0.840	0.856	0.932	0.918	0.940	0.929	0.928
TE(CBOW)	0.736	0.744	0.740	0.734	0.846	0.797	0.821	0.891	0.951	0.948	0.950	0.950
TE(Skip)	0.737	0.747	0.742	0.735	0.852	0.788	0.819	0.887	0.893	0.900	0.896	0.896
TE(GloVe)	0.721	0.729	0.725	0.719	0.841	0.776	0.807	0.884	0.869	0.879	0.874	0.874
TR+TE(CBOW)	0.788	0.793	**0.791**	**0.785**	0.881	0.854	0.867	0.933	**0.954**	**0.958**	**0.956**	**0.956**
TR+TE(Skip)	0.787	0.794	**0.791**	0.784	**0.887**	**0.857**	**0.871**	0.936	0.942	0.953	0.947	0.947
TR+TE(GloVe)	**0.789**	0.794	**0.791**	**0.785**	0.877	**0.857**	0.867	**0.937**	0.936	0.938	0.937	0.937

In contrast, EETM acquires content semantics based on the word co-occurrence patterns and transforms word embeddings to topic embeddings by utilizing topic distribution. Thus, it is able to filter trivial context patterns of word embeddings and to produce effective topic embeddings.

We observe from Table 4 that TR+TE performs better than TR and TE individually. Note text representations of topics and word embeddings are based on different but complementary text patterns. EETM can not only retain the original characteristics of topics and embeddings, but also make them share their semantic information as much as possible, leading to a better text representation method. While LFTM and TWE perform well in combining topics and embeddings, evidenced by good performance in our experiments, they fail to take the difference between topics and embeddings into consideration, limiting the further improvement of their performance.

LDA+WE outperforms LDA or WE individually, which also indicates that topics and embeddings are complementary. For 20NG and Reuters, their category labels are highly related to topic information of the corpus, thus LDA+WE produces similar results that are closer to LDA. In Hotel, the sentiment labels rely more on the details of content, which makes LDA+WE closer to WE. However, the concatenation of LDA and WE does not deal with their relationship, which limits the further improvement of the performance.

Finally, another phenomenon can be seen from Table 4 is that F1 values of all compared methods on 20NG and Hotel are close to their accuracy values, although on Reuters, F1 is apparently lower than accuracy. This is because the categories of 20NG and Hotel are balanced, while the categories of Reuters are imbalanced. The results show our EETM can work well on both balanced and imbalanced data sets.

5.4.2 Clustering Task Evaluation

We now evaluate EETM on text clustering task, one of important applications in text analysis domain. We adopt state-of-the-art spectral clustering[14] implemented in *scikit-learn* library as our clustering method for EETM, as it is suitable to non-flat geometry case, which is suitable for text contents. Widely used cosine similarity is utilized to measure the semantic relationship between two documents, and it is applied to all the EETM methods and baselines in our experiments. Furthermore, EETM could acquire more abundant information of documents to measure the semantic between two documents. By making use of the topic distribution and topic embeddings of a document, we define an additional *topic embedding similarity measurement* (denoted as **TES** in Table 5) between document d_i and d_j for EETM, which is formulated as follows.

$$TES(d_i, d_j) = \sum_{k=1}^{K} \frac{2\theta_{ik}\theta_{jk}}{\theta_{ik} + \theta_{jk}} Cosine(\boldsymbol{t}_{ik}, \boldsymbol{t}_{jk}) \tag{27}$$

The value of $TES(d_i, d_j)$ varies in the closed interval $[-1, 1]$. When $i = j$, $TES(d_i, d_j)$ reaches the maximum value 1. $TES(d_i, d_j)$ can be regarded as weighted-averaged cosine similarity of topic embeddings, which simultaneously takes the topic distribution and topic embeddings into consideration. Clearly, documents that have similar topic distribution and context patterns will get higher similarities.

To evaluate the performance of clusters, we employ the averaged entropy of categories in each cluster as the evaluation metric [44]. Lower value of the averaged entropy indicates that the cluster covers documents from fewer and consistent categories, which means better performance. Especially, when a cluster only covers documents from one category, its entropy is 0, i.e. the best quality. The number of clusters is set to 40 initially, and increase 40 each time until 200. The detailed results of clustering evaluation are listed in Table 6 (Appendix D), where best performance is listed in

14. http://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html

TABLE 5
Best performance of clustering task.

Method	20NG	Reuters	Hotel
BOW	3.308	0.888	0.793
LDA	2.427	0.640	0.550
GaussianLDA	3.165	1.115	0.950
LFTM	2.607	0.792	0.570
Doc2Vec	3.438	1.297	0.696
TWE	2.242	0.627	0.462
WE(CBOW)	2.342	0.761	0.537
WE(Skip)	2.269	0.682	0.525
WE(GloVe)	2.298	0.687	0.578
LDA+WE(CBOW)	2.050	0.550	0.657
LDA+WE(Skip)	**2.029**	0.557	0.616
LDA+WE(Glove)	2.042	0.574	0.657
TR(CBOW)	2.179	**0.525**	0.683
TR(Skip)	2.419	0.593	0.452
TR(GloVe)	2.341	0.624	0.462
TE(CBOW)	3.334	1.222	0.545
TE(Skip)	3.361	1.219	0.718
TE(GloVe)	3.363	1.231	0.747
TR+TE(CBOW)	2.263	0.603	0.529
TR+TE(Skip)	2.481	0.614	**0.417**
TR+TE(GloVe)	2.330	0.701	0.452
TES(CBOW)	2.222	0.628	0.631
TES(Skip)	2.215	0.658	0.519
TES(GloVe)	2.211	0.711	0.520

bold font. For the sake of clarity, we listed best performance of each method in table 5, and compared the best results of TR, TR+TE, and TES with LDA and TWE in Figure 2, as they are generally better than other compared methods and baselines.

From Figure 2, it can be confirmed that the increasing number of clusters will generally reduce the averaged entropy of categories in each cluster. The TES(Glove) works generally better on 20NG, while TR(CBOW) works better on Ruters, and TR+TE(Skip) gets the best performance on Hotel, as shown in Figure 2. The content complexity of 20NG is higher than Reuters and much higher than Hotel. Thus, 20NG is more difficult to be clustered properly. Our defined TSE measurement can leverage more semantic information of documents to distinguish complicated content, leading to better performance in 20NG.

LFTM also utilizes word embeddings to enhance the topic representation. However, it fails to make use of the difference between embeddings and topic-word relations, leading to its worse performance than LDA. TWE uses the topic-word distributions to form refined topic embeddings, so it outperforms LDA. TR is the topic distribution learned by EETM, which benefits from both topic-word distributions and embeddings by utilizing the proposed information integration strategy. As 20NG and Reuters are topic category data sets, their contents are highly depend on the quality of topic distribution. Thus, TR outperforms LDA and LFTM. TR also works better than TWE on 20NG and Reuters data, and obtains similar good results with TWE on Hotel data.

Topic embeddings learned by EETM focuses on the semantic details of documents, which are filtered by the topic-word distribution. Semantics of text are complex, and the performance of different data set is closely related to the content of documents. TE focuses on the specific topical contents of each document, leading to less similarity of embedding vectors from different documents. TR also takes in context information carried by word embeddings, which may weaken the topic information. Thus, for data set whose labels strongly rely on topic information and overall context information, such as 20NG, LDA+WE would provide the best performance. In task such as the sentiment clustering task on Hotel data, the sentiment clusters are highly associated with the details of each topic, and especially topics with sentiment information are essential. Thus TR experiences performance improvements by combining with TE, to achieve the best performance on TR+TE(Skip). As LDA+WE fails to discover sentiment topics, its performance is surely limited in the sentiment clustering task. With EETM, multi-perspective text representation can be obtained to deal with different text analysis tasks.

BOW model produces competitive results in classification tasks, but it suffers from data sparsity in clustering tasks. We observe that GaussianLDA and Doc2Vec are not suitable for clustering tasks too and their results are worse than BOW.

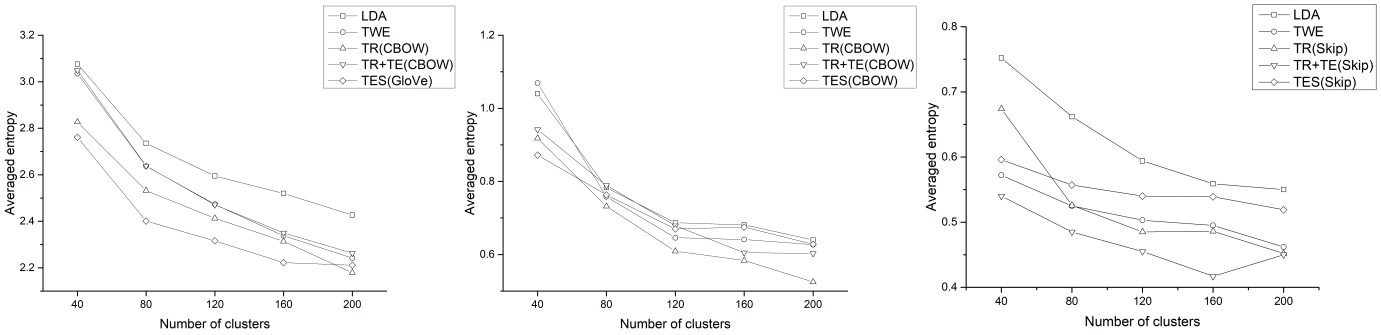
In summary, the results of text and sentiment clustering tasks on three data sets demonstrate that EETM produces the best representation to encode the text semantics in documents, and our proposed information integration strategy can take advantages of both topics and word embeddings.

5.5 Discussion

In our experiments, we have compared EETM with baselines on two common text analysis tasks. During the parameters selection and classification tasks, 5-fold cross validation is used to verify the results. In the qualitative assessment of topic coherence, EETM could assign proper keywords into related topics even on small or domain-specified data sets. In the clustering tasks, topic distribution and topic embedding can be combined for different data sets. Furthermore, our topic embedding similarity measurement based on EETM is suitable for 20NG. The results of our experiments demonstrated that the proposed information integration strategy and EETM could work well in both supervised and unsupervised text representation tasks.

The motivation of our integration framework is to map the semantic information of topics into corresponding semantic embedding structure. Both topic embeddings and topic distributions can receive improvements from this strategy. The different aspects of a document are described by topic distribution and embeddings together, by using which we can design more accurate measurements for representing the text content. Besides, it also allows topics and topic embeddings to maintain their own characteristics. The final text representation of EETM can be established by combining these two parts systematically. Thus, EETM could build comprehensive text representation that contains both topic information and structure information learned by word embeddings. The topic embedding matrix and topic distributions for a single document can further be used to construct refined semantic measurements.

EETM is a hybrid model, and the model setting of its components is essential to EETM. There are two major settings for EETM, namely the topic number K, and the pre-trained word embeddings. The topic number K of EETM



(a) 20NG (b) Reuters (c) Hotel

Fig. 2. Performance comparison on text clustering task. Best performance of TR, TR+RE, and TES on three pre-trained word embedding methods are compared with LDA and TWE.

has the same function to traditional topic models. Thus, it can be determined by conventional tricks such as cross-validation or empirical values. The quality of the pre-trained word embeddings is essential to EETM as well. EETM does not update the word embeddings during the learning procedure. The meaning of a topic particularly depends on the content of the document, which can be treated as weighted average of its related word embeddings. Updating word embeddings with the topic embeddings tends to average the word embeddings as well. Thus, word embeddings are not being updated in EETM. The pre-trained word embeddings can be selected by testing on certain data sets and tasks.

Based on our experiments, similar improvements have been observed on EETM with three pre-trained word embedding methods. EETM is not sensitive to the dimensionality of pre-trained word embeddings. For different data sets and tasks, the performance of EETM with different pre-trained word embeddings has witnessed marginal differences, i.e. EETM works quite stably with different pre-trained word embeddings.

6 Conclusions

In this paper, we have proposed a novel integration strategy to construct an Embedding Enhanced Topic Model (EETM), facilitating two different perspectives of text patterns, from topic models and word embeddings, can be integrated effectively.

Our experimental results have demonstrated that EETM works very well on three major text analysis tasks, namely text classification, sentiment classification, text clustering, across three different benchmark data sets, indicating EETM is able to produce high quality text representations. In addition, original characteristics of topic models and word embeddings can be retained to maintain their individual unique structure and functional advantages for better text comprehension. Moving forward, we believe EETM can potentially be used in other tasks, such as information retrieval, similarity analysis etc. Furthermore, the proposed information integration framework has potential applications in various scenarios, such as combining different types of machine learning models, integrating different representation methods etc.

Acknowledgments

The authors would like to thank the editors and all anonymous reviewers for their valuable comments and suggestions which have significantly improved the quality and presentation of this paper. This work was supported by the National Natural Science Foundation of China (NSFC nos.61632011, 61573231, 61672331, 61432011).

References

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] W. Zhao, Z. Guan, L. Chen, X. He, D. Cai, B. Wang, and Q. Wang, "Weakly-supervised deep embedding for product review sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2017.

[3] H. Gui, J. Liu, F. Tao, M. Jiang, B. Norick, L. Kaplan, and J. Han, "Embedding learning with events in heterogeneous information networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2428–2441, 2017.

[4] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, "Sse: Semantically smooth embedding for knowledge graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 884–897, 2017.

[5] S.-J. Shin and I. C. Moon, "Guided htm: Hierarchical topic model with dirichlet forest priors," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 2, pp. 330–343, 2017.

[6] Y. Zhuang, H. Wang, J. Xiao, F. Wu, Y. Yang, W. Lu, and Z. Zhang, "Bag-of-discriminative-words (bodw) representation via topic modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 977–990, 2017.

[7] Z. Hai, G. Cong, K. Chang, P. Cheng, and C. Miao, "Analyzing sentiments in one go: A supervised joint topic modeling approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1172–1185, 2017.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[9] S. Li, J. Zhu, and C. Miao, "A generative word embedding model and its low rank positive semidefinite solution," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 1599–1609.

[10] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015.

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *Proceedings of Empiricial Methods in Natural Language Processing(EMNLP)*, vol. 14. Association for Computational Linguistics, 2014, pp. 1532–43.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of NIPS*, 2013, pp. 3111–3119.

[13] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1137–1155, 2003.
[14] Z. Cao, S. Li, Y. Liu, W. Li, and H. Ji, "A novel neural topic model and its supervised extension." in *AAAI*, 2015, pp. 2210–2216.
[15] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 299–313, 2015.
[16] S. Li, T.-S. Chua, J. Zhu, and C. Miao, "Generative topic embedding: a continuous representation of documents," in *Proceedings of The 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 666–675.
[17] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings." in *AAAI*, 2015, pp. 2418–2424.
[18] J. Law, H. H. Zhuo, J. He, and E. Rong, "Ltsg: Latent topical skip-gram for mutually learning topic model and vector representations," *arXiv preprint arXiv:1702.07117*, 2017.
[19] R. Das, M. Zaheer, and C. Dyer, "Gaussian lda for topic models with word embeddings," in *Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2015, pp. 795–804.
[20] G. Xun, Y. Li, W. X. Zhao, J. Gao, and A. Zhang, "A correlated topic model using word embeddings," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence.[doi¿ 10.24963/ijcai.2017/588]*, 2017, pp. 4207–4213.
[21] H. Xu, M. Dong, D. Zhu, A. Kotov, A. I. Carcone, and S. Naar-King, "Text classification with topic-based word embedding and convolutional neural networks," in *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2016, pp. 88–97.
[22] S. Zheng, H. Bao, J. Xu, Y. Hao, Z. Qi, and H. Hao, "A bidirectional hierarchical skip-gram model for text topic embedding," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 855–862.
[23] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in Neural Information Processing Systems*, 2008, pp. 121–128.
[24] P. Zhang, S. Wang, and D. Li, "Cross-lingual sentiment classification: Similarity discovery plus training data adjustment," *Knowledge-Based Systems*, vol. 107, pp. 129 – 141, 2016.
[25] P. Zhang, H. Gu, M. Gartrell, T. Lu, D. Yang, X. Ding, and N. Gu, "Group-based latent dirichlet allocation (group-lda): Effective audience detection for books in online social media," *Knowledge-Based Systems*, vol. 105, pp. 134 – 146, 2016.
[26] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum, "Integrating topics and syntax," in *Advances in Neural Information Processing Systems*, 2004, pp. 537–544.
[27] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 487–494.
[28] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 248–256.
[29] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in Neural Information Processing Systems*, 2009, pp. 1081–1088.
[30] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 384–394.
[31] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.
[32] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars." in *Proceedings of ACL*, 2013, pp. 455–465.
[33] D. Tang, "Sentiment-specific representation learning for document-level sentiment analysis," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 2015, pp. 447–452.
[34] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck, "Contextual lstm (clstm) models for large scale nlp tasks," *arXiv preprint arXiv:1602.06291*, 2016.
[35] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.
[36] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," *arXiv preprint arXiv:1507.07998*, 2015.
[37] Y. Ji and J. Eisenstein, "Representation learning for text-level discourse parsing." in *Proceedings of ACL*, 2014, pp. 13–24.
[38] W. Zhang, Q. Yuan, J. Han, and J. Wang, "Collaborative multi-level embedding learning from reviews for rating prediction," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, 2016, pp. 2986–2992.
[39] J. He, Z. Hu, T. Berg-Kirkpatrick, Y. Huang, and E. P. Xing, "Efficient correlated topic modeling with topic embedding," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. New York, NY, USA: ACM, 2017, pp. 225–233. [Online]. Available: http://doi.acm.org/10.1145/3097983.3098074
[40] L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin, "Discriminative neural sentence modeling by tree-based convolution," *arXiv preprint arXiv:1504.01106*, 2015.
[41] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.
[42] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.
[43] J. Han Lau, D. Newman, and T. Baldwin, "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality," in *14th Conference of the European Chapter of the Association for Computational Linguistics 2014, EACL 2014*, 01 2014, pp. 530–539.
[44] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

Peng Zhang received the PhD degree from Shanxi University. He is a lecturer in the School of Information, Shanxi University of Finance and Economics. His research interests include machine learning, computational natural language understanding, and text sentiment analysis.

Suge Wang received the PhD degree from Shanghai University. She is a professor in the School of Computer and Information Technology, Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University. Her research interests include natural language processing, text sentiment analysis, and machine learning. She has published more than 30 articles in international journals.

Deyu Li received the PhD degree from Xian Jiaotong University. He is a professor in the School of Computer and Information Technology, Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University. His research interests include artificial intelligence, granular computing, data mining, and machine learning. He has published more than 60 articles in international journals.

Xiaoli Li is currently a department head and principal scientist at the Institute for Infocomm Research, A*STAR, Singapore. He also holds adjunct professor positions at Nanyang Technological University. His research interests include data mining, machine learning, AI, and bioinformatics. He has been serving as a (senior) PC member/workshop chair/session chair in leading data mining and AI related conferences (including KDD, ICDM, SDM, PKDD/ECML, WWW, IJCAI, AAAI, ACL and CIKM). Xiao-Li has published more than 180 high quality papers and won numerous best paper/benchmark competition awards.

Zhikang Xu received the MS degree from Shanxi University. He is working toward the PhD degree in the School of Computer Engineering and Science, Shanghai University. His research interests include machine learning and multi-label classification.