A New Effective Neural Variational Model with Mixture-of-Gaussians Prior for Text Clustering

Miao Li[1,3], Hongyin Tang[1,3], Beihong Jin[1,3] and Chengqing Zong[2,3]
[1]*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China*
[2]*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[3]*University of Chinese Academy of Sciences, Beijing, China*

***Abstract*—Text clustering is one of the fundamental tasks in natural language processing (NLP) and text data mining. It remains challenging because texts have complex internal structure besides the sparsity in the high-dimensional representation. In the paper, we propose a new Neural Variational model with mixture-of-Gaussians prior for Text Clustering (abbr. NVTC) to reveal the underlying textual manifold structure and cluster documents effectively. NVTC is a deep latent variable model built on the basis of the neural variational inference. In NVTC, the stochastic latent variable, which is modeled as one obeying a Gaussian mixture distribution, plays an important role in establishing the association of documents and document labels. On the other hand, by joint learning, NVTC simultaneously learns text encoded representations and cluster assignments. Experimental results demonstrate that NVTC is able to learn clustering-friendly representations of texts. It significantly outperforms several baselines including VAE+GMM, VaDE, LCK-NFC, GSDPMM and LDA on four benchmark text datasets in terms of *ACC*, *NMI*, and *AMI*. Furthermore, NVTC learns effective latent embeddings of texts which are interpretable by topics of texts, where each dimension of latent embeddings corresponds to a specific topic.**

***Index Terms*—Text clustering, Deep generative model, Neural variational inference, Latent variable model**

I. Introduction

Text clustering [1] refers to the process of analyzing a group of texts and classifying similar texts into the same category based on their content and inherent topics. As one of the most fundamental tasks in natural language processing and text data mining, text clustering has been widely applied in news summarization, document organization and browsing, and content recommendation on social websites. The sparse high-dimensional representation space of texts [3] brings the difficulty to text clustering. This problem reflects the fact that any lexicon is rather large while each document contains only a small number of words. Besides high-dimensional sparsity, text data have complex internal structures, which has been the obstacle for effective text clustering.

Different from text stream clustering, hierarchical text clustering, and co-clustering [7], we focus on document-level partitional text clustering, in which the structure of the clustering output is flat.

In the past decade, a large family of text clustering methods, e.g., GSDPMM [3], a text clustering method based on the Dirichlet Process Multinomial Mixture model, have been proposed. With the development of deep learning technology, deep neural networks have been used to model texts due to their inherent capabilities of highly non-linear transformations. Deep neural networks are expected to extract underlying complex features of texts and map raw texts to clustering-friendly representations to improve the quality of text clustering results. The conventional approaches which leverage neural networks for text clustering are building neural networks to extract effective features of texts, and treat the feature transformation and the clustering as two independent processes. In these text clustering methods, the assumptions in dimensionality reduction and feature transformation are generally independent of the assumptions required by the clustering techniques. Thus, there is no theoretical guarantee that the neural network would learn feasible representations [27]. In recent years, the neural variational inference [23] has attracted considerable attention. Deep latent variable models which combine the composability of graphical models with the flexible modeling capabilities of deep networks are expected to clustering texts effectively if properly optimized [20]. As a kind of the most popular deep generative models, Variational Autoencoders (VAEs) [22] have exposed their abilities of extracting underlying complex structure of texts. Based on VAEs, VaDE [5], a deep clustering framework is presented. VaDE shows good results in clustering images. But it does not exhibit good performance in text clustering and lacks interpretability for the learned latent embeddings of texts.

In this paper, we propose a new effective Neural Variational model with mixture-of-Gaussians for Text Clustering named NVTC. NVTC is a deep latent variable model which is on the basis of the neural variational inference. NVTC focuses on clustering texts and its latent variable is interpretable via text topics. Specifically, we treat the latent variable as the bridge between documents and their labels, which is reflected explicitly in the design of the loss function of NVTC. On this point, NVTC comes away with a different strategy from VaDE, where VaDE simplifies the factorization of the variational posterior probability according to the mean-field theory so that the connection information between the latent variable and document labels is dropped.

In summary, the main contributions of this paper can be concluded as follows:

- NVTC, a new effective neural variational model with a mixture-of-Gaussians prior for text clustering is pro-

This work has been supported by National Key R&D Program of China (No. 2017YFC0803300)

posed. Based on the neural variational inference, the loss function of NVTC is derived from a new factorization of the variational posterior with considering the positive connection between documents and their category labels.
- By optimizing both the text encoding and clustering tasks and learning latent code and clustering assignment jointly, NVTC can simultaneously cluster texts effectively and acquire latent coding of texts in a continuous lowdimensional space. These latent representations can be applied to downstream tasks, e.g., text indexing and document similarity computation. Furthermore, the softmax decoder of NVTC makes the learned latent embedding interpretable by text topics.
- Extensive experiments on four benchmark datasets are conducted to evaluate the text clustering performance of NVTC. Experimental results demonstrate that NVTC outperforms several baselines, i.e., VAE+GMM, VaDE, LCK-NFC, GSDPMM and LDA, in terms of three of the most widely used unsupervised clustering metrics (i.e., *ACC*, *NMI*, *AMI*). Specifically, NVTC significantly outperforms GSDPMM by approximately 0.15, 0.04 and 0.05 in *ACC*, *NMI* and *AMI* on 20NG, respectively. Moreover, NVTC gets a much lower perplexity than NVDM, VaDE and LDA on all benchmark datasets.

II. Related Work

Existing text clustering methods can be classified into two categories, traditional statistical methods and neural network-based methods. The former can be further divided into similarity-based text clustering [11] and model-based text clustering [10].

In general, similarity-based methods use the vector space model to represent text data, then define and calculate the similarities among them, and finally run clustering algorithms (e.g., K-means [12], DBSCAN [13], BIRCH [14] and their typical variants) on the similarity matrices. Clustering models which are based on the matrix decomposition [16] [17] and the spectral graph analysis [15] have also achieved good results in text clustering. For example, [17] proposes a new decomposition method by maximizing the correntropy between the original matrix and the product of two factorized low-rank matrices for document clustering. [11] proposes a new approach to extract the document concepts which are consistent with the local geometry of document submanifold and then treat the documents with similar concepts as a cluster, leveraging the power of both CF (Concept Factorization) and graph Laplacian regularization. In these methods, since texts are represented by high-dimensional sparse vectors, feature extraction and dimension reduction are keys to effective text clustering.

Model-based text clustering methods assume that texts are generated by a statistical model, and apply the Expectation-Maximization algorithm or the Gibbs sampling method to estimate the parameters. For example, [3] [10] and [21] employ the DPMM (the Dirichlet Process Multinomial Mixture model, a nonparametric mixture model based on the Dirichlet process) to model the text generating process. Especially, the model presented in [3], i.e., GSDPMM, models documents through a multinomial distribution on bag-of-words representations. On the other hand, model-based topic modeling (such as LDA [18]) also can be used for clustering of plain texts by assigning each document to the most probable topic.

In recent years, some work focuses on learning effective text features by deep neural networks to improve clustering performance [27]. For example, [19] encodes text features in a self-taught manner by a convolutional neural network, and then performs text clustering by K-means algorithm. [20] proposes a denoising auto-encoder for dimension reduction and clusters texts in the learned latent space.

In the neural network-based models, some work accomplishes both feature learning and clustering assignments simultaneously [6] [5] [28] [27]. These previous studies have shown that optimizing these two tasks jointly can improve their performance significantly which can uncover the real underlying structure of text data. For example, for text data, LCK-NFC [26] treats feature extraction and clustering as a united process, where clustering results can be used as feedback information to optimize the network parameters. However, LCK-NFC only optimizes the clustering loss [27] and has the risk of learning corrupted text representations. Especially, the LCK-NFC cannot show good text clustering performance.

We notice that VAE-based deep clustering models [27] can learn features and cluster assignments simultaneously, and the most important is that these models combine the probability graph models with neural networks.

VAE can be viewed as a simplified version of the neural variational inference framework [23]. This framework in [23] is for text modeling and combines the variational Bayesian approach with the flexibility and extendibility of neural networks. In this framework, an encoder-decoder structure is responsible for extracting a latent embedding for each input. Specifically, a random latent variable $\mathbf{z}$ is assumed to be generated from a Gaussian distribution and $\mathbf{z}$ is inferred by a variational distribution $q(\mathbf{z}|\mathbf{x})$ which is parameterized by an encoder network where $\mathbf{x}$ is the input text data. It is worth noting that this variational distribution is an approximation to the real posterior distribution $p(\mathbf{z}|\mathbf{x})$. With the latent variable $\mathbf{z}$ as input, the decoder reconstructs the original input data $\mathbf{x}$. The parameters of this model are obtained by maximizing the variational lower bound of the log-likelihood $\log p(\mathbf{x})$. The goal of this objective function is to reduce the reconstruction error and make the variational distribution $q(\mathbf{z}|\mathbf{x})$ close to the prior $p(\mathbf{z})$. This neural variational inference framework has achieved success in many tasks with different data, including images [25] and texts.

Existing VAE-based deep clustering models, e.g., VaDE [5] and GMVAE [28] are mixture models. In detail, both VaDE and GMVAE select the Gaussian mixture distribution to describe the clustering structure of the data, and replace the isotopic Gaussian of the general VAEs with the Gaussian mixture distribution as the prior distribution of the latent

variables. Due to setting the assumption that the data are generated from a Gaussian mixture distribution, the cluster of the data is equal to the component in the mixture distribution which the latent variables are generated from. GMVAE is more complex than the VaDE model, and experimental results show that the clustering performance of GMVAE is not as good as that of VaDE [27]. Among the above models, some do not aim at text data, and the left cannot obtain good performance of text clustering, and no models can make the latent variables interpretable.

III. NVTC Model

In this section, we first introduce the detail derivation process of the loss function of NVTC in the view of likelihood maximization, and then describe the implementation of the model under the encoder-decoder network structure in the framework of neural variational inference.

A. Loss function

NVTC is an unsupervised generative text clustering model. From the view of the neural variational inference framework, the model is obtained by replacing the Gaussian distribution of the latent variable in VAEs with the Gaussian mixture distribution, designing a new loss function and optimizing the evidence lower bound of the newly derived log-likelihood loss function. In detail, NVTC adopts a Gaussian mixture distribution as the prior to describe the clustering structure of text data, discarding the isotropic Gaussian distribution which is regarded as the prior distribution of the latent variable in vanilla VAEs. Further, the parameters of NVTC are learned by maximizing the log-likelihood of input texts. Under the assumption that documents are generated from a mixture of Gaussians, inferring a cluster assignment of a document turns into inferring which component of the latent distribution the document is generated from. Thus, after maximizing the evidence lower bound, the clustering assignments can be inferred directly by the learned GMM (Gaussian Mixture Model).

Let $\mathbf{x} \in \mathbb{R}^{|V|}$ be the bag-of-words representation of a document, V and $|V|$ denote the vocabulary and its size, respectively. Let the discrete variable $y \in \{1, 2, ..., k\}$ denote the clustering assignment of the document and k represents the number of document clusters.

We introduce a continuous latent variable $\mathbf{z} \in \mathbb{R}^d$, which is generated from a Gaussian mixture distribution, and let $\mathbf{z}$ represent a latent embedding of a document in the latent space, which learns salient and clustering-friendly features of the document. Specifically, this latent embedding implies the category of the document. That is, $p(\mathbf{z}) = \sum_y p(\mathbf{z}|y)p(y)$.

Next, we give the generative process of documents in the form of probabilistic graphical model in NVTC, as shown in Fig. 1. First we choose a category y from $p(y)$, and then sample a latent variable $\mathbf{z} \sim p(\mathbf{z}|y) = \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\sigma}_y^2)$ from the current category y, which represents the embedding of the document in the latent space, and finally generate $\mathbf{x}$ from $p(\mathbf{x}|\mathbf{z})$.

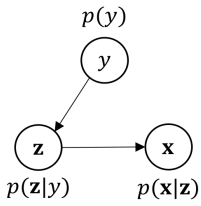


Fig. 1. The graphical model of the generative process of documents in NVTC.

According to the above generative process, the joint probability $p(\mathbf{x}, \mathbf{z}, y)$ of NVTC can be factorized to $p(\mathbf{x}|\mathbf{z})p(\mathbf{z}|y)p(y)$.

Since the log-likelihood of documents which is represented by $\log p(\mathbf{x})$ is intractable, in order to derive its approximation, we apply the variational Bayes method to obtain the evidence lower bound (i.e., ELBO) and define the ELBO as the loss function of NVTC (i.e., $\mathcal{L}_{NVTC}$). Eq. 1 reveals the key points.

$$\begin{aligned}\log p(\mathbf{x}) &= \log \int \sum_y p(\mathbf{x}, \mathbf{z}, y) d\mathbf{z} \\ &\geq \mathbb{E}_{q(\mathbf{z},y|\mathbf{x})} \left[\log p(\mathbf{x}, y, \mathbf{z})\right] - \mathbb{E}_{q(\mathbf{z},y|\mathbf{x})} \left[\log q(\mathbf{z}, y|\mathbf{x})\right] \\ &= \mathcal{L}_{NVTC}\end{aligned} \tag{1}$$

where $q(\mathbf{z}, y|\mathbf{x})$ is the variational distribution which is an approximation to the posterior distribution $p(\mathbf{z}, y|\mathbf{x})$.

According to the generative process described in Fig. 1, we have

$$\begin{aligned} p(\mathbf{x}|\mathbf{z}, y) &= p(\mathbf{x}|\mathbf{z}), \\ p(\mathbf{z}, y|\mathbf{x}) &= p(\mathbf{z}|\mathbf{x})p(y|\mathbf{z}, \mathbf{x}) = p(\mathbf{z}|\mathbf{x})p(y|\mathbf{z}) \end{aligned} \tag{2}$$

Because $q(\mathbf{z}, y|\mathbf{x})$ is an approximation of the true posterior $p(\mathbf{z}, y|\mathbf{x})$, $q(\mathbf{z}, y|\mathbf{x})$ is factorized to $q(y|\mathbf{z})q(\mathbf{z}|\mathbf{x})$ according to Eq. 2. Then, $\mathcal{L}_{NVTC}$ can be derived as follows.

$$\begin{aligned}&\mathcal{L}_{NVTC} = \\ &\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z})\right] + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(y|\mathbf{z})} \left[\log p(\mathbf{z}|y)\right] \\ &\qquad + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(y|\mathbf{z})} \left[\log p(y)\right] - \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(y|\mathbf{z})} \left[\log q(\mathbf{z}|\mathbf{x})q(y|\mathbf{z})\right] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}) - \mathbb{E}_{q(y|\mathbf{z})} \log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|y)} - KL\left(q(y|\mathbf{z})||p(y)\right)\right]\end{aligned} \tag{3}$$

where $\mathcal{L}_{NVTC}$ incorporates the network loss and the clustering loss. Specifically, the first term of $\mathcal{L}_{NVTC}$ implies the expectation of smaller reconstruction error (i.e., the network loss) when given the latent variable $\mathbf{z}$ sampled from the variational distribution, and ensures that more salient features of texts can be extracted. The second term indicates that partitioning $\mathbf{z}$ into different clusters and the substantial latent variable $\mathbf{z}$ can be generated from the inferred category y, which ensures that $\mathbf{z}$ can be assigned to the real cluster. The third term is to regularize the variational distribution of y to approach its prior distribution.

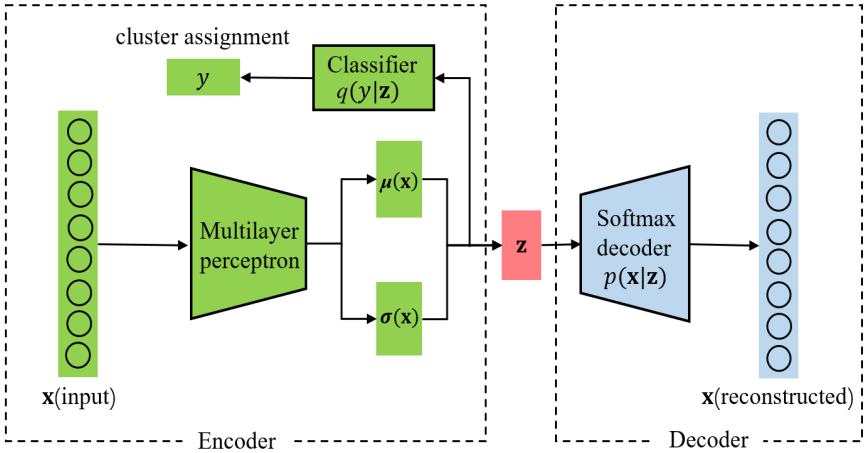


Fig. 2. The implementation of NVTC in an encoder-decoder structure.

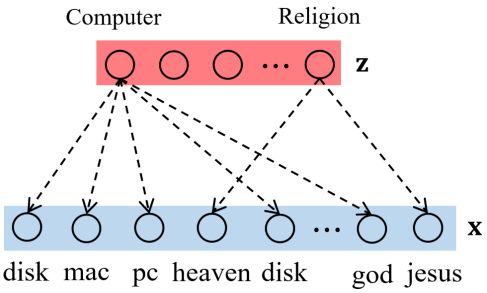


Fig. 3. The structure of the softmax decoder.

B. Neural inference network

After the derivation of the loss function, we build a neural inference network in an encoder-decoder structure to infer text clustering. In the light of the basic encoder-decoder structure, the encoder takes a document $\mathbf{x}$ as input and produces the latent variable $\mathbf{z}$, and then $\mathbf{z}$ is sent to the decoder which reconstructs the original $\mathbf{x}$. To perform the text clustering, the target encoder-decoder network should be able to produce the values needed by the loss function calculation and has an extra classification neural network for the latent variable to obtain the cluster assignment of $\mathbf{x}$.

Fig. 2 shows the overall structure of our neural inference network. The following give the details of our network.

In NVTC, we have assumed that the variational posterior distribution $q(\mathbf{z}|\mathbf{x})$, i.e., the approximation of the true posterior, is a Gaussian distribution, that is, $q(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}^2(\mathbf{x}))$. Therefore, first, we take document $\mathbf{x}$ in a bag-of-words representation as input, employ a multilayer perceptron which is composed of three fully connected layers as the encoder to obtain the mean $\boldsymbol{\mu}(\mathbf{x})$ and standard deviation $\boldsymbol{\sigma}(\mathbf{x})$ of above variational posterior Gaussian distribution. In order to obtain the latent variable $\mathbf{z}$ and reduce the variance in stochastic estimation, we sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and reparameterize $\mathbf{z} = \boldsymbol{\mu}(\mathbf{x}) + \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}(\mathbf{x})$ from the above variational posterior Gaussian distribution [24].

Then, in order to decode $\mathbf{z}$ to the original document, we assume that each word in the document is generated independently, i.e., $p(\mathbf{x}|\mathbf{z}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\mathbf{z})$, where N is the number of the words in the document and $\mathbf{x}_i \in [0,1]^{|V|}$ is the one-hot representation of the word at position i. Thus, $\mathcal{L}_{NVTC}$ can be further derived as

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\sum_{i=1}^{N} \log p(\mathbf{x}_i|\mathbf{z}) - \mathbb{E}_{q(y|\mathbf{z})} \log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|y)} - KL\left(q(y|\mathbf{z})||p(y)\right)\right] \tag{4}$$

Meanwhile, for the conditional probability of word $\mathbf{x}_i$, we adopt a multinomial logistic regression to predict it. That is,

$$p(\mathbf{x}_i|\mathbf{z}) = \frac{\exp\{\mathbf{z}^T\mathbf{W}\mathbf{x}_i + \mathbf{b}_{\mathbf{x}_i}\}}{\sum_{j=1}^{|V|} \exp\{\mathbf{z}^T\mathbf{W}\mathbf{x}_j + \mathbf{b}_{\mathbf{x}_j}\}} \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{d \times |V|}$ is the linear transformation from the latent variable space to the output space and it learns the semantic embeddings of words, and $\mathbf{b}_{\mathbf{x_i}}$ is a bias term. $\mathbf{W}$ and $\mathbf{b}_{\mathbf{x_i}}$ are the parameters which can be learned by a densely connected softmax decoder. Employing a softmax layer as the decoder also enables the latent variable interpretable by topics. Fig. 3 demonstrates the linear structure of softmax decoder. Empirically, the latent variable can be interpreted by document topics and each dimension of $\mathbf{z}$ corresponds to a specific topic. Each topic can be represented by the words with top-n highest weights in the row of $\mathbf{W}$. After having $\mathbf{W}$ and $\mathbf{b}_{\mathbf{x_i}}$, the first term in the loss function is able to be calculated.

Finally, it is also assumed that $p(\mathbf{z}|y)$ is a Gaussian distribution, that is, each component of $p(\mathbf{z})$ is $\mathcal{N}(\boldsymbol{\mu}_y, \mathbf{I})$. Considering that y is discrete, we set $p(y)$ as a discrete uniform distribution which $q(y|\mathbf{z})$ approaches in the third term in the loss function, that is, we expect the number of documents in each category to be roughly equal. Further, considering that $q(y|\mathbf{z})$ in $\mathcal{L}_{NVTC}$ can be any type of classifiers that are suitable for the latent variable, we also use a multilayer perceptron composed of two fully connected layers as the classifier to get the cluster assignment y with $\mathbf{z}$ as input. Thus, we can calculate the second and the third term in the loss function.

In the training stage, the expressive power of $\mathbf{z}$ and the clustering performance is improved simultaneously by optimizing the objective function $\mathcal{L}_{NVTC}$. Thus, after training, we can obtain the cluster assignments of the documents by feeding the learned latent embeddings into the classifier $q(y|\mathbf{z})$ directly.

IV. Experiments

In the experimental study, we evaluate the clustering performance of NVTC on four benchmark text datasets and conduct comprehensive comparisons with several text clustering baselines (i.e., VAE+GMM, VaDE, LCK-NFC, GSDPMM and LDA). Three widely used external clustering metrics including *ACC* (Unsupervised Clustering Accuracy) [5] [6], *NMI* (Normalized Mutual Information) [8] and *AMI* (Adjusted Mutual Information) [9] are adopted. Moreover, we evaluate the capability of NVTC for document modeling by calculating the perplexity (*PPL*) [23] of texts. We visualize the learned latent embeddings of texts, which show the difference of discriminative quality of the latent representations. We also interpret the latent embeddings by topics of texts. Experimental results show that NVTC achieves the state-of-the-art text clustering performance on all the experimental benchmark datasets. Furthermore, we analyze the impact of parameter k in NVTC on the text clustering performance.

A. Experimental setup

We use four benchmark text datasets which are widely used in evaluation of text clustering and classification methods, including *20NewsGroups*[1] (20NG), *Reuters RCV1-v2*[2] (Reuters), Reuters-16 [2] and *Yahoo Answer* (Yahoo)[4]. 20NG is a collection of newsgroup documents, consisting of 18,846 with the average length of 137.85 words from 20 major newsgroups. Reuters is a large collection from Reuters newswire stories with around 810,000 English news labeled with a category tree in original Reuters dataset. To make a direct comparison with [5], we choose four root categories (i.e., corporate/industrial, government/social, markets, and economics) and finally get 685,071 documents by discarding all documents with multiple labels. In order to conduct experiment on diversified text datasets, we adapt Reuters to be another large-scale dataset, named Reuters-16, consisting of 563,335 documents from 16 third-level categories in the category tree. Yahoo is also a large-scale text dataset, containing millions of questions and answers from ten topics including Society&Culture, Science&mathematics, etc. The summary statistics of these datasets are shown in Table I after preprocessing (i.e., converting all letters into lowercase, removing stop words and stemming).

Training details of NVTC are as follows. In the encoder, we set the neuron numbers of the three-layer perceptron as 512, 1024 and 512, respectively. The neuron numbers of two full connected layers in the classifier are set to 256 and k, respectively. To make a direct and reasonable experimental comparison, we set the hyper-parameter k to the number of ground-truth categories of text datasets, which is further analyzed in Section IV-D. To facilitate the training of NVTC, the learning rate α of Adam optimizer [30] is selected from [2e-3, 1e-3] and β_1 is set from [0.5, 0.9]. Practically, we notice that when α=1e-3 and β_1=0.9, NVTC achieves the best performance. In order to improve the performance and stability of the model, we employ a batch normalization layer before the input of the decoder, i.e., conducting batch normalization on the reparametrization results. We apply a batch size of 32 to train NVTC model with 50 and 200 dimensional latent embeddings respectively. Our fully connected layers are initialized with random initialization. Moreover, we apply Adam optimizer to learn the parameters of networks in NVTC and adopt ReLU (Rectified Linear Unit) as the activation function in the ecoder and classifier. To eliminate the impact of randomness on experiments, we report the average results of all metrics after running each model with different settings for twenty independent trials on each dataset.

TABLE I
Summary statistics for 20NG, Reuters, Reuters-16 and Yahoo.

Dataset	#Documents	#Clusters	#Vocabulary	#Avg_Len
20NG	18,846	20	181,754	137.85
Reuters	685,071	4	268,063	119.65
Reuters-16	563,335	16	216,485	108.08
Yahoo	1,460,000	10	752,604	40.42

B. Unsupervised text clustering results

In the experiments for evaluation of text clustering performance, we make quantitative comparison among NVTC and other baseline models, including VAE+GMM, VaDE, LCK-NFC, GSDPMM and LDA. VAE+GMM refers to treat text feature learning and clustering as a pipeline process. That is, latent embeddings of texts in the latent space are learned by VAE, then the static latent embeddings are fed into GMM for clustering. To make a direct comparison, in VAE+GMM, the network structures of encoder and decode are set the same as those of NVTC and the number of components of GMM is set to the number of the ground-truth categories of text dataset. For LDA, we treat the topics found by LDA as clusters and assign each document to the cluster with the highest value in its topic proportion vector. Following [3], we set the number of topics in LDA to k, and set $\alpha = 0.002k$, $\beta = 0.1$. Furthermore, we reproduce VaDE, LCK-NFC and GSDPMM, all of which follow the optimal settings in their original papers. Each of these models is run for twenty times to obtain their average performance on different experimental datasets.

Table II shows the clustering results of NVTC on these four text datasets, compared with five baseline models in terms of three widely-used clustering metrics on four different experimental datasets. It is obvious that NVTC significantly outperforms other baseline models in terms of *ACC*, *NMI* and *AMI*.

Compared with current state-of-the-art performance that GSDPMM reports, NVTC outperforms GSDPMM by 0.147, 0.033 and 0.051 respectively in terms of *ACC*, *NMI*, and *AMI*

[1]http://qwone.com/jason/20Newsgroups
[2]http://trec.nist.gov/data/reuters/reuters.html

TABLE II
CLUSTERING PERFORMANCE (*ACC*, *NMI*, *AMI*) OF NVTC WITH A COMPARISON TO FIVE BASELINE MODELS INCLUDING VAE+GMM, VADE, LCK-NFC, GSDPMM AND LDA ON FOUR DATASETS.

	Metrics	NVTC	VAE+GMM	VaDE	LCK-NFC	GSDPMM	LDA
20NG	*ACC*	**0.7247**	0.5365	0.5678	0.5321	0.5776	0.5503
	NMI	**0.7011**	0.6427	0.6322	0.6278	*0.667	*0.602
	AMI	**0.6889**	0.5966	0.6175	0.5531	0.6375	0.5422
Reuters	*ACC*	**0.8132**	0.6289	0.7938	0.6077	0.7133	0.6033
	NMI	**0.5654**	0.4753	0.5021	0.4306	0.4935	0.4132
	AMI	**0.5534**	0.4697	0.4969	0.4297	0.4921	0.4013
Reuters-16	*ACC*	**0.5877**	0.5103	0.5233	0.4319	0.5177	0.4001
	NMI	**0.5115**	0.4721	0.4832	0.4499	0.4739	0.4067
	AMI	**0.4357**	0.4067	0.4132	0.3677	0.4099	0.3586
Yahoo	*ACC*	**0.6220**	0.5667	0.5745	0.5679	0.5724	0.5633
	NMI	**0.5170**	0.4339	0.4651	0.4196	0.4725	0.4123
	AMI	**0.4951**	0.4198	0.4576	0.4088	0.4652	0.3969

* is from the original paper of GSDPMM.

on 20NG. It is because that NVTC, as an instantiation of deep latent variable models and a variant of VAEs, has a stronger ability to extract complex internal structure of texts than traditional generative models.

From Table II, we can also find that the performance of NVTC is much better than VAE+GMM and LDA. Obviously, NVTC learns clustering-friendly text representations, which enhances the performance of text clustering. While comparing NVTC with VAE+GMM or LDA, their major difference is that NVTC jointly optimizes the feature extraction and clustering. This indicates that it is more effective to simultaneously learn text features and cluster documents, which verifies the observations in [27].

The performance of NVTC is also much better than LCK-NFC, though the latter also jointly extracts text features and learns cluster assignments. This is because that NVTC models the generative process of documents via a deep generative model, which incorporates the network loss and the clustering loss and has the inherent superiority in extracting complex features of texts, while LCK-NFC only optimizes the clustering loss which is calculated by the Silhouette Coefficient in its training process. LCK-NFC has the risk of learning corrupted text representations and it is hard for it to reveal the complex structure of texts.

Experimental results also show that NVTC is better than VaDE at text clustering. This is because these two models have different approximations of the posterior in the derivation loss function. Specifically, VaDE approaches the variational posterior $q(\mathbf{z}, y|\mathbf{x})$ with $q(y|\mathbf{x})q(\mathbf{z}|\mathbf{x})$ according to the mean-field theory, and NVTC factorizes $q(\mathbf{z}, y|\mathbf{x})$ to $q(y|\mathbf{z})q(\mathbf{z}|\mathbf{x})$ based on the real generative process of texts. Actually, there really exists a relation b etween $\mathbf{z}$ and y when conditioning on $\mathbf{x}$. Although VaDE takes some tricks to mitigate the information loss caused by the assumption of the mean-field distribution, it really has a negative impact on text clustering results. In addition to better text clustering performance, NVTC is easy to train, because it dose not need to pretrain the internal encoder and decoder by executing other additional models, however in VaDE, networks need to be pretrained by stacked auto-encoders.

Moreover, NVTC can simultaneously get very good performance in all the three metrics on 20NG, while *ACCs* of other baseline models are far lower than their *NMIs*, as shown in the first two rows of Table II. Among all the text datasets, NVTC gets the worst performance on Yahoo because it has more noise data and even supervised classification models cannot perform well on them. However, NVTC gets better performance than other models on Yahoo, which shows that it has stronger ability of clustering texts.

TABLE III
PPLs OF NVTC, NVDM, VADE AND LDA ON 20NG AND REUTERS. LATENT VARIABLES ARE SET TO HAVE 50 AND 200 DIMENSIONS.

	NVTC		NVDM		VaDE		LDA	
	50d	200d	50d	200d	50d	200d	50d	200d
20NG	**687**	701	796	830	778	821	996	971
Reuters	**486**	509	542	521	531	546	1367	1124

C. Document modeling results

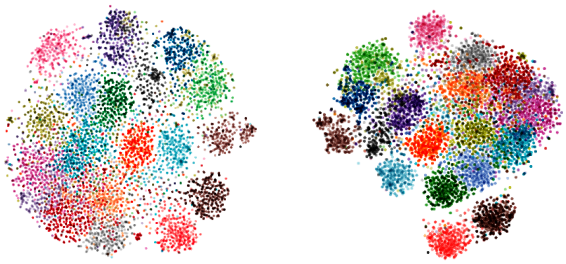
As a by-product of the effective text clustering, NVTC can learn the latent embedding of each document effectively in the unsupervised setting. These latent embeddings can be used in a number of downstream text mining tasks. Here, the latent embedding represents one code of the original document in the latent space. We compare the capability of document modeling of NVTC with that of VaDE, LDA and NVDM [23] which is a instantiation of VAEs for documents, giving up GSDPMM and LCK-NFC because they cannot output effective embeddings of documents. Specifically, we make the encoder and decoder structure of NVDM the same as the encoder and decoder of NVTC to make ensure a fair comparison. Following [23], we evaluate the capability of text modeling by using the variational lower bound to approximate *PPLs* on texts datasets.

As shown in Table III, the results show that the *PPLs* of NVTC are lower than any other models in both text datasets, which demonstrates that NVTC has a stronger capability of document modeling than any other models.

In order to analyze the internal structure of the embeddings learned by these models, we visualize the latent embeddings by mapping the 50-dimensional embeddings on 20NG dataset into two-dimensional vectors using t-SNE [29] and project them in a two-dimensional plane. Fig. 4 shows the projections

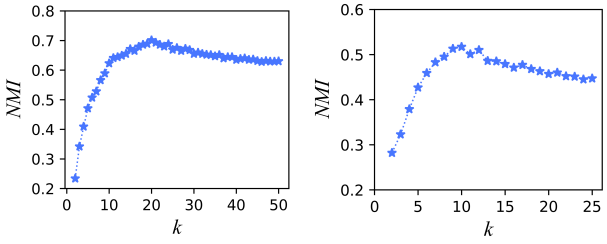
TABLE IV
SEMANTIC INTERPRETATION OF THE LATENT VARIABLE. TOP-10 WORDS ARE THE WORDS WITH TOP-10 HIGHEST WEIGHTS. THE FIRST ROW REFERS TO THE DEDUCED TOPICS.

Topic	*Computer*	*University*	*Email*	*Space*	*Religion*	*Medical*
	card	berkeley	file	earth	god	health
	window	duke	list	planet	christian	medical
	mac	purdue	mail	nasa	jesus	doctor
	pc	princeton	ftp	moon	church	disease
Top-10	driver	yale	send	shuttle	church	patient
words	monitor	stanford	addresss	launch	bible	drug
	disk	colorado	data	satellite	christ	treatment
	software	mit	email	station	love	medicine
	memory	geogia	package	flight	heaven	cancer
	modem	harvard	message	orbit	faith	gordon



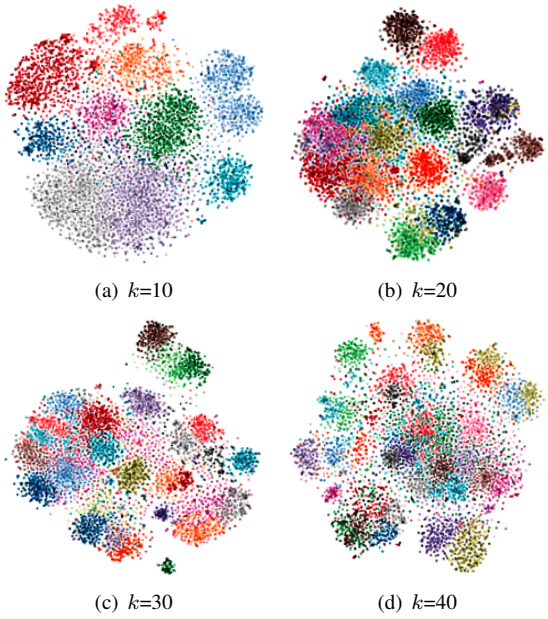
(a) (b)

Fig. 4. Visualization of 50-dimensional latent embeddings using t-SNE. Different colors correspond to different ground-truth category labels. (a) is the projection of latent embeddings learned by NVDM on the 20NG dataset. (b) is the projection of latent embeddings learned by NVTC on the 20NG dataset.



(a) 20NG (b) Yahoo

Fig. 5. *NMIs* of NVTC for clustering on 20NG and Yahoo when k varies.



(a) k=10 (b) k=20

(c) k=30 (d) k=40

Fig. 6. Projections of latent embeddings learned by NVTC on two-dimensional plane when k is set to different values.

of latent embeddings learned by NVDM and NVTC, where each dot represents a document and the color of the dot indicates its ground-truth category label. It is obvious that the embeddings learned by NVTC indicate a better discriminative quality since the embeddings in different categories are separated more clearly and the embeddings in the same categories are gathered more closely.

The reason why NVTC has a stronger capability of document modeling than NVDM is that assumptions of their latent variable priors are different. NVDM assumes that the latent variable is generated from a Gaussian distribution. From this perspective, NVDM is an instantiation of VAEs. However, NVTC assumes that the prior distribution of the latent variable is a mixture of Gaussians which makes latent embeddings more clustering-friendly.

Besides, we observe that latent embeddings learned by NVTC can also be interpreted by the topics of texts. As mentioned in Section III-B, each row of $\mathbf{W} \in \mathbb{R}^{d\times|V|}$ is analogous to the topic-word distribution of the topic model, and the element values of $\mathbf{W}$ denote the probabilities of words. Thus, each dimension of the embedding is corresponding to a specific topic. As same as studies in the regular topic models, the semantic interpretation of each topic can be represented by the words with top-n highest values in the row of $\mathbf{W}$. Table IV shows the semantic interpretation of the 50-dimensional latent embeddings learned by NVTC on 20NG. We sample 6 topics randomly and show the top-10 representative words in the second row of Table IV. According to these representative words, we can deduce the relative topics such as *Computer*, *University* and etc., which are consistent with the topics in the first row of Table IV. However, it is worth noting that this kind of topical interpretability of latent embeddings has not been explicitly analyzed by other state-of-the-art text clustering models, including GSDPMM and VaDE.

D. The analysis of hyper-parameter k

In the previous comparison experiments, k is given as a hyper-parameter and set to the number of ground-truth categories of texts. To analyze the impact of different k on the clustering performance of NVTC, we set k to different values and conduct experiments on 20NG and Yahoo datasets. Figs. 5(a) and 5(b) show *NMI* values of the 20NG and Yahoo datasets when k varies. As we can see, NVTC obtains the highest *NMI* on 20NG when k is set to 20, and obtains the highest *NMI* on Yahoo when k is set to 10. These two numbers are exactly matched with the numbers of ground-truth categories of 20NG and Yahoo, respectively.

Fig. 6 shows the projections of latent embeddings learned by NVTC on the 20NG dataset under different k, i.e., 10, 20, 30, and 40. Different colors represent different clustering assignments. Obviously, NVTC can cluster documents effectively under different k. Specifically, when $k = 20$, the boundaries of clusters are more discriminative, which indicates that these latent embeddings are more effective to express the information of the ground-truth categories of text datasets.

V. Conclusion

We propose a new and more effective text clustering model NVTC with a mixture-of-Gaussians prior based on the neural variational inference. Experimental results show that NVTC significantly outperforms several baseline models including VAE+GMM, GSDPMM, VaDE, LDA and LCK-NFC in terms of *ACC*, *NMI* and *AMI*, and achieves the state-of-the-art text clustering performance on all the experimental benchmark datasets. Besides the more effective clustering performance, NVTC learns latent embeddings of texts in a lower-dimensional continuous latent space. Each dimension of the latent embeddings is in one-to-one correspondence with the topic of the text, so that NVTC has certain semantic interpretability. Furthermore, NVTC clustering results can be used for text storing and retrieval, and the learned latent embeddings of texts can be used in many downstream text mining tasks, such as text similarity calculation. As the following research, we will incorporate more textual features (e.g., sequential information) into NVTC and introduce pre-trained word embeddings into NVTC to further improve the performance of text clustering.

References

[1] C. C. Aggarwal, and C. Zhai, "A survey of text clustering algorithms," *Mining text data*, Springer, pp. 77-128, 2012.

[2] D. D. Lewis ,Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *Journal of machine learning research* vol. pp. 361-397, 2004.

[3] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," in *32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 625-636.

[4] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems (NIPS)*, 2015, pp. 649-657.

[5] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," in IJCAI, 2017, pp. 1965-1972.

[6] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for cluste ring analysis," in *International conference on machine learning*. MIT Press, 2016, pp. 478-487.

[7] C. C. Aggarwal, "Machine Learning for Text," Springer, pp. 79-82, 2018.

[8] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2761-2773, 2010.

[9] N. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837-2854, 2010.

[10] G. Yu, R. Huang, and Z. Wang, "Document clustering via dirichlet process mixture model with feature selection," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 763-772.

[11] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 902-913, 2011.

[12] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1969, pp. 281-297.

[13] M. Ester, H. P. Kriegel, S. Jorg, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*. ACM, 1996, pp. 226-231.

[14] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *ACM SIGMOD*. ACM, 1996, pp. 103-114.

[15] D. Cai, X. He, J. Han, "SRDA: An efficient algorithm for large-scale discriminant analysis," *IEEE transactions on knowledge and data engineering*, vol. 20, no. 1, pp. 1-12, 2008.

[16] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Information Processing & Management*. vol. 42, no. 2, pp. 373-386, 2006.

[17] L. Li, J. Yang, Y. Xu, Z. Qin, and H. Zhang, "Documents clustering based on max-correntropy nonnegative matrix factorization," in *International conference on machine learning and cybernetics*. IEEE, 2014, pp.850-855.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of machine Learning research*, vol. 3, pp. 993-1022, 2003.

[19] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, and H. Hao, "Short text clustering via convolutional neural networks," in *Proceedings of NAACL-HLT*. ACL, 2015, pp.62-69.

[20] M. Leyli-Abadi, L. Labiod, and M. Nadif, "Denoising autoencoder as an effective dimensionality reduction and clustering of text data," in *Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference (PAKDD)*. Springer, 2017, pp. 801-813.

[21] L. Hu, J. Li, X. Li, C. Shao, and X. Wang, "TSDPMM: Incorporating prior topic knowledge into Dirichlet process mixture models for text clustering," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. ACL, 2015, pp. 878-792.

[22] Y. Kim, S. Wiseman, A. M. Rush, "A Tutorial on Deep Latent Variable Models of Natural Language," *arXiv preprint* arXiv:1812.06834, 2018.

[23] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *International conference on machine learning (ICML)*. MIT Press, 2016, pp.1727-1736.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations (ICLR)*, 2014.

[25] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in neural information processing systems (NIPS)*, 2016, pp. 2352-2360.

[26] Y. Fan, G. Liu, M. Kui, and S. Zhaoying, "Neural Feedback Text Clustering with BiLSTM-CNN-Kmeans," *IEEE Access*, vol. 6, pp. 57460-57469, 2018.

[27] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39501-39514, 2018.

[28] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee *et al.*, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv preprint*, arXiv:1611.02648, 2016.

[29] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, pp. 2579-2605, 2008.

[30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.