Neural Sparse Topical Coding

Anonymous ACL submission

Abstract

Topic models with sparsity enhancement have been proven to be effective at learning discriminative and coherent latent topics of short texts, which is critical to many scientific and engineering applications. However, the extensions of these models require carefully tailored graphical models and re-deduced inference algorithms, limiting their variations and applications. We propose a novel sparsity-enhanced topic model, Neural Sparse Topical Coding (NSTC) base on a sparsity-enhanced topic model called Sparse Topical Coding (STC). It focuses on replacing the complex inference process with the back propagation, which makes the model easy to explore extensions. Moreover, the external semantic information of words in word embeddings are incorporated to improve the representation of short texts. To illustrate the flexibility offered by the neural network based framework, we present three extensions based on NSTC without re-deduced inference algorithms. Experiments on Web Snippet and 20Newsgroups datasets demonstrate that our models outperform existing methods.

1 Introduction

Topic models with sparsity enhancement have proven to be effective tools for exploratory analysis of the overload of short text content. The latent representations learned by these models are central to many applications. However, these models have trouble to rapidly explore variations for the approximate inference methods of them. Even subtle variations on models can increase the complexity of the inference methods, which is especially apparent for non-conjugate models.

With the development of deep learning, many works combine topic models with neural language model to overcome the computation complexity of topic models (Larochelle and Lauly, 2012a; Cao et al., 2015; Tian et al., 2016). Most of these methods adopt multiple neural network layers to model the generation process of each document. Nevertheless, these methods yield the same poor performance in short texts as traditional topic models. There are also many works introducing new techniques such as word embeddings to traditional topic models. Word embeddings has proven to be effective at capturing syntactic and semantic information of words. Many works (Das et al., 2015; Hu and Tsujii, 2016; Xun et al., 2017) have shown that the additional semantics in word embeddings can enhance the performance of traditional topic models. Yet these models have the same trouble in making extensions as traditional topic models.

Based on the above observations, we propose Neural Sparse Topical Coding (NSTC) by jointly utilizing word embeddings and neural network with a sparsity-enhanced topic model, Sparse Topical Coding (STC). We adopt neural network to model the generation process of STC to simplify the complex inference and improve flexibility, and incorporate external semantics provided by word embeddings to improve the overall accuracy. To illustrate the dramatic flexibility offered by the end-to-end neural network, we present three extensions based on NSTC. Our proposed models all benefit from both sides: 1) when compared with the neural based topic models, which stuck in the sparseness of word co-occurrence information, they show how the sparsity mechanism and word embeddings enrich the features and improve the performance; 2) while with topic models with sparsity enhancement, our models present how the black-box inference method of neural network accelerates the training process and increases the flexibility. The main contributions of this paper are summarized as follows:

1. We develop a sparsity-enhanced neural topic model NSTC to learn sparse representations of words and documents and introduce the general word semantic information by word embeddings.

2. We present three variants of NSTC to illustrate the great flexibility of our framework.

3. We evaluate the effectiveness and efficiency of our models by conducting experiments on 20 Newsgroups and Web Snippet datasets.

2 Related Work

Topic models with sparsity enhancement: The performance of traditional topic models are compromised by the sparse word co-occurrence information when applied in short texts. To overcome the bottleneck, there have been many efforts to address the problem of sparsity in short texts. Based on traditional LDA, (Williamson et al., 2010) introduced a *Spike and Slab* prior to model the sparsity in finite and infinite latent topic structures of text. To consider the dual-sparsity of topics per document and terms per topic, (Lin et al., 2014) proposed a dual-sparse topic model that addresses the sparsity in both the topic mixtures and the word usage. There are also some non-probabilistic sparse topic models, which can directly control the sparsity by imposing regularizers. For example, the non-negative matrix factorization (NMF) (Heiler and Schnörr, 2006) formalized topic modeling as a problem of minimizing loss function regularized by lasso. Similarly, (Zhu and Xing, 2011) presented sparse topical coding (STC) by utilizing the Laplacian prior to directly control the sparsity of inferred representations. Additionally, (Peng et al., 2016) presented sparse topical coding with sparse groups (STCSG) to find sparse word, topic and document representations of texts. However, over complicated inference procedure of these sparse topic models make them difficult to rapidly explore variations.

Topic models with word embeddings: There are many works tried to incorporate word embeddings with topic models to improve the performance. (Das et al., 2015) proposed a new technique for topic modeling by treating the document as a collection of word embeddings and topics itself as multivariate Gaussian distributions in the embedding space. However, the assumption that topics are unimodal in the embedding space is not appropriate, since topically related words can occur distantly from each other in the embedding space. Therefore, (Hu and Tsujii, 2016) proposed latent concept topic model (LCTM), which modeled a topic as a distribution of concepts, where each concept defined another distribution of word vectors. (Nguyen et al., 2015) proposed Latent Feature Topic Modeling (LFTM), which extended LDA to incorporate word embeddings as latent features. Lately, (Xun et al., 2017) proposed a novel correlated topic model using word embeddings, which is enable to exploit the additional word-level correlation information in word embeddings and directly model topic correlation in the continuous word embedding space. However, these models also have trouble to rapidly explore variations.

Neural Topic Models: There are also some efforts trying to combine topic models with neural networks to represent words and documents simultaneously. (Larochelle and Lauly, 2012a) proposed a neural network topic model that is similarly inspired by the Replicated Softmax. (Cao et al., 2015) proposed a novel neural topic model (NTM) where the representation of words and documents are efficiently and naturally combined into a uniform framework. (Das et al., 2015) proposed a new technique for topic modeling by treating the document as a collection of word embeddings and topics itself as multivariate Gaussian distributions in the embedding space. To address the limitations of the bag-of-words assumption, (Tian et al., 2016) proposed Sentence Level Recurrent Topic Model (SLRTM) by using a Recurrent Neural Networks (RNN) based framework to model long range dependencies between words. Nevertheless, most of aforementioned works yield poor performance in short texts.

3 Neural Sparse Topical Coding

Firstly, we define that $D = \{1, ..., M\}$ is a document set with size M, $T = \{1, ..., K\}$ is a topic collection with K topics, $V = \{1, .., N\}$ is a vocabulary with N words, and $w_d = \{w_{d,1}, ..., w_{d,|I|}\}$ is a vector of terms representing a document d, where I is the index of words in document d, and $w_{d,n} (n \in I)$ is the frequency of word n in document d. Moreover, we denote $\beta \in \mathbb{R}^{NK}$ as a topic dictionary for the whole document set with k bases, $\theta_d \in \mathbb{R}^K$ is the document code of document d and $s_{d,n} \in \mathbb{R}^K$ is the word code of word n in document d. To yield interpretable patterns, (θ, s, β) are constrained to be non-negative.

3.1 Sparse Topical Coding

STC is a hierarchical non-negative matrix factorization for learning hierarchical latent representations of input samples. In STC, each document and each word is represented as a low-dimensional code in topic space, which can be employed in many tasks. Based on the global topic dictionary β of all documents with K topic bases sampled from a uniform distribution, the generative process of each document d is described as follows:

1. Sample the document code θ_d from a prior $p(\theta_d) \sim Laplace(\lambda^{-1})$.
2. For each observed word n in document d:
   (a) Sample the word code $s_{d,n}$ from a conditional distribution $p(s_{d,n}|\theta_d) \sim supergaussian(\theta_d, \gamma^{-1}, \rho^{-1})$.
   (b) Sample the observed word count $w_{d,n}$ from a distribution $p(w_{d,n}|s_{d,n}. * \beta_n) \sim Poisson(s_{d,n}. * \beta_n)$

To achieve sparse word codes, STC defines $p(s_{d,n}|\theta_d)$ as a product of two component distributions $p(s_{d,n}|\theta_d) \sim p(s_{d,n}|\theta_d, \gamma)p(s_{d,n}|\rho)$, where $p(s_{d,n}|\theta_d, \gamma)$ is an isotropic Gaussian distribution, and $p(s_{d,n}|\rho)$ is a Laplace distribution. The composite distribution is super-Gaussian: $p(s_{d,n}|\theta_d) \propto exp(\gamma||s_{d,n}\theta_d||_2^2\rho||s_{d,n}||_1)$. With the Laplace term, the composite distribution tends to yield sparse word codes. For the same purpose, the prior distribution $p(\theta_d)$ of document codes is a Laplace prior. Although STC has closed form coordinate descent equations for parameters (θ, s, β), it is inflexible for its complex inference process.

3.2 Neural Network View of Sparse Topical Coding

We devote to rebuild STC with a neural network to simplify it's inference process via BackPropogation. After generating the topic dictionary from neural network, our model follows the generative story below for each document d:

1. For each word n in document d:
   (a) Sample a latent variable word code $s_{d,n} \sim f_g(d, n)$.
   (b) Sample the observed word count $w_{d,n}$ from $p(w_{d,n}|s_{d,n}, \beta_n) \sim Poisson(s_{d,n}. * \beta_n)$

In our model, we have several assumptions:

1) To simplify our model and acclerate the inference process, we collapse the document code from our model. As illuatrated in (Bai et al., 2013) and STC paper (Zhu and Xing, 2011), we can naturally generate each document code via a aggregation of all sampled word codes among all topics, after inferring the global topic dictionary and the word codes of words belong to each document:

$$\theta_d = \sum_{d=1}^{D}\sum_{n=1}^{N_d} s_{d,nk}\,\beta_{kn} / \sum_{d=1}^{D}\sum_{n=1}^{N_d}\sum_{k=1}^{K} s_{d,nk}\,\beta_{kn};$$

2) We replace the composite super-Gaussian prior of the word codes and the uniform distribution of the topic dictionary with the neural network. In the topic dictionary neural network, we introduce the word semantic information via word embeddings to enrich the feature space for short texts;

3) Similar to STC, the observed word count is sampled from Poisson distribution, which is more appropriate for the discrete count data than other exponential family distributions.

3.3 Neural Sparse Topical Coding

In this section, we introduce the detailed layer structures of NSTC in Figure 1. We explicitly explain each layer of NSTC below:

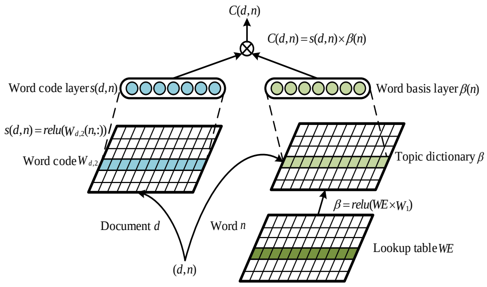


Figure 1: Schematic overview of *NSTC*.

Input layer (n, d): A word n of document $d \in D$, where D is a document set.

Word embedding layer $(WE \in \mathbb{R}^{N\times 300})$: Supposing the word number of the vocabulary is N, this layer devotes to transform each word to a distributed embedding representation. Here, we adopt the pre-trained embeddings by GloVe based on a large Wikipedia dataset [1].

Word code layers ($s \in \mathbb{R}^{N \times K}$): These layers generate the K-dimensional word code of input word n in document d.

$$s(d, n) = f_s(d, n) \tag{1}$$

where f_s is a multilayer perceptron. In order to achieve interpretable word codes as in STC, we constrain s to be non-negative, therefore we apply the relu activation function on the output of the neural network. Although imposing non-negativity constraints could potentially result in sparser and more interpretable patterns, we exert l_1 norm regularization on s to further keep the sparse assumption.

Topic dictionary layers ($\beta \in \mathbb{R}^{N \times K}$): These layers aims at converting WE to a topic dictionary similar to the one in STC.

$$\beta(n) = f_\beta(WE) \tag{2}$$

where f_β is a multilayer perceptron. We make a simplex projection among the output of topic dictionary neural network.We normalize each column of the dictionary via the simplex projection as follow:

$$\beta_{.k} = project(\beta_{.k}), \forall k \tag{3}$$

The simplex projection is the same as the sparsemax activation function in (Martins and Astudillo, 2016), providing the theoretical base of its employment in a neural network trained with backpropagation. After the simplex projection, each column of the topic dictionary is promised to be sparse, non-negative and united.

Score layer ($C \in \mathbb{R}^{1 \times |I_d|}$): NSTC outputs the matching score of a word n and a document d with the dot product of $s(d, n)$ and $\beta(n)$ in this layer. The output score is utilized to approximate the observed word count $w_{d,n}$.

$$C(d, n) = s(d, n). * \beta(n) \tag{4}$$

Given the count $w_{d,n}$ of word n in document d, we can directly use it to supervise the training process. According to the architecture of our model, for each word n and each document d, the cost function is:

$$L = l(w_{d,n}, C(d, n)) + \lambda ||s_{d,n}||_1 \tag{5}$$

where l is the log-Poisson loss, λ is the regularization factors.

3.4 Extensions of NSTC

To future illustrate the benefits of a black box inference method, which allows rapidly explore new models, we present three variants of NSTC.

Deep l_1 Approximation. STC is based on the idea of sparse coding, in which the sparse code s of the input w can be obtained by solving the loss function for a given dictionary β. In (Gregor and LeCun, 2010), the parameterized encoder, named learned ISTA (LISTA) was proposed to efficiently approximate the l_1 based sparse code. Based on the consideration, we present a enhanced NSTC via employing the deep l_1 regularized encoder similar to LISTA, named NSTCE. We devise a feed-forward neural network as illustrated in Figure 2, to efficiently approximate the l_1 based sparse code s of the input w.

$$F(w_d; W_d, b_d) = relu(w_d * W_d + b_d) \tag{6}$$

The goal is to make the prediction of neural network predictor F after the fixed depth as close as possible to the optimal set of coefficients s^* in Eq.4. To jointly optimizing all parameters with the dictionary β together, we add another term to the loss function in Eq.4, and enforce the representation s to be as close as possible to the feed forward prediction (Kavukcuoglu et al., 2010):

$$\begin{aligned} L = & l(w_{d,n}, C(d, n)) + \lambda ||s_{d,n}||_1 \\ & + \alpha \sum_n ||s_d - F(w_d; W_d, b_d)||_2^2 \end{aligned} \tag{7}$$

Minimizing the loss with respect to s produces a sparse representation that simultaneously reconstructs the word count and is not too different from the predicted representation.

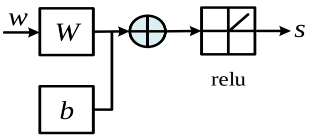


Figure 2: Deep l_1 encoder.

Group Sparse Regularization. Based on STC, (Bai et al., 2013) presented GSTC to discover document-level sparse or admixture proportion for short texts, in which the group sparse is employed to achieve sparse code at document level by taking into account the structure of bag of words. Here,

[1] http://nlp.stanford.edu/projects/glove/

we just need to add the group sparse regularization on the weight matrix, to make a neural network extension of GSTC, called NGSTC. We consider each column of s_d as a group.

$$L = l(w_{d,n}, C(d,n)) + \lambda \sum_{k=1}^{K} ||s_{d,.k}||_2 \tag{8}$$

Sparse Group Lasso. Similar to GSTC, STCSG (Peng et al., 2016) was proposed to learn sparse word and document codes, which relaxes the normalization constraint of the inferred representations with sparse group lasso. Base on STCSG, we propose a novel neural topic model called NSTCSG. We impose the sparse group lasso on the word code, and have the following cost function:

$$L = l(w_{d,n}, C(d,n)) + \lambda_1 ||s_{d,n}||_1 + \lambda_2 \sum_{k=1}^{K} ||s_{d,.k}||_2 \tag{9}$$

These three models have the same neural network structures as NSTC, and can be trained end to end with out re-deduced mathematical inference. Moreover, group and sparse group sparsity can help reduce the intrinsic complexity of the model by eliminating neurons as shown in Figure 3, and thus can help obtain practical speed ups in deep neural networks.

3.5 Optimization

For the first two models with the lasso regularizer, we can directly ulitize the end to end stochastic gradient descent (SGD) to perform optimizing. The last two optimizing objectives of NGSTC and NSTCSG are formed as a combination of both smooth and non-smooth terms, they can be solved via proximal stochastic gradient descent. The proximal gradient algorithm first obtains the intermediate solution via SGD on the loss only, and then optimize for the regularization term via performing Euclidean projection of it to the solution space, as in the following formulation:

$$\min_{s_{d,n}^{t+1}} R(s_{d,n}^{t+1}) + \frac{1}{2}||s_{d,n}^{t+1} - s_{d,n}^{t+\frac{1}{2}}||_2^2 \tag{10}$$

where R is the regularization, $s_{d,n}^{t+\frac{1}{2}}$ the intermediate solution obtained by SGD, $s_{d,n}^{t+1}$ is the variable to obtain after the current iteration. For the group lasso, the above problem has the closed-form solution. The proximal operator for the group lasso regularizer in Eq.8 is given as follow:

$$prox_{GL}(s_{d,nk}) = (1 - \frac{\lambda}{||s_{d,.k}||_2})_+ s_{d,nk} \tag{11}$$

The proximal operator for the sparse group lasso regularizer in Eq.9 is given as follow:

$$\begin{aligned} prox_{SGL}(s_{d,nk}) =& (1 - \frac{\lambda_2}{||sign(s_{d,.k}, \lambda_1)||_2})_+ \\ & sign(s_{d,nk}, \lambda_1) \end{aligned} \tag{12}$$

The detailed algorithm framework of NGSTC and NSTCSG is shown in Algorithm 1.

Algorithm 1 Training Algorithm for our models

```
Require: a document $d \in D$
1: $t = t + 1$
2: repeat
3:     Compute the partial derivatives of weight matrices,$s$, and $\beta$ with a non-regularized objective;
4:     Update weight matrices, $s$, and $\beta$ using SGD.
5:     Update $s$ using proximal operator
6:     Update $\beta$ using simplex projection.
7: until convergence
```

4 Experiments

4.1 Data and Setting

We perform our experiments on two benchmark datasets:

- **20Newsgroups**: is comprised of 18775 newsgroup articles with 20 categories, and contains 60698 unique words [2].
- **Web Snippet**: includes 12340 Web search snippets with 8 categories, we remove the words with fewer than 3 characters and with document frequency less than 3 in the dataset.[3].

We compare the performance of the NSTC with the following baselines:

- **LDA** (Blei et al., 2001). A classical probabilistic topic model. We use the LDA package[4] for its implementation. We use the settings with iteration number $n = 2000$, the

[2] http://www.qwone.com/ jason/20Newsgroups/

[3] http://jwebpro.sourceforge.net/data-web-snippets.tar.gz

[4] https://pypi.python.org/pypi/lda

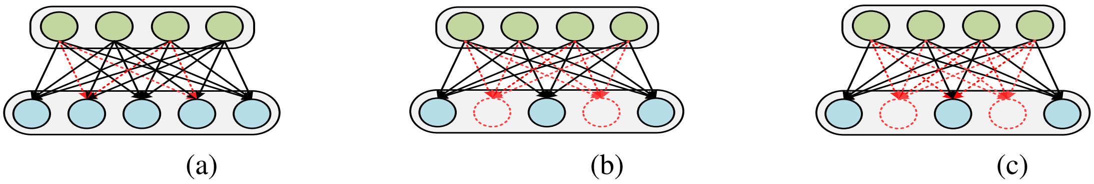


Figure 3: (a) Lasso: the Lasso penalty removes elements without optimizing neuron-level considerations (highlighted in red). (b) Group lasso: when grouping weights from the the same input neuron into each group, the group sparsity has an effect of completely removing some neurons (highlighted in red). (c) Sparse group lasso: it combines the advantages of the former two formulations, which can remove some neurons and elements (highlighted in red).

Dirichlet parameter for distribution over topics $\alpha = 0.1$ and the Dirichlet parameter for distribution over words $\eta = 0.01$.

- **STC** (Zhu and Xing, 2011). It is a sparsity-enhanced non-probabilistic topic model. We use the code released by the authors [5]. We set the regularization constants as $\lambda = 0.3, \rho = 0.0001$ and the maximum number of iterations of hierarchical sparse coding, dictionary learning as 100.
- **DocNADE** (Larochelle and Lauly, 2012b). An unsupervised neural network topic model of documents and have shown that it is a competitive model both as a generative model and as a document representation learning algorithm [6]. In DocNADE, the hidden size is 50, the learning rate is 0.0004 , the bath size is 64 and the max training number is 50000.
- **GaussianLDA** (Das et al., 2015). A new technique for topic modeling by treating the document as a collection of word embeddings and topics itself as multivariate Gaussian distributions in the embedding space [7]. We use default values for the parameters.

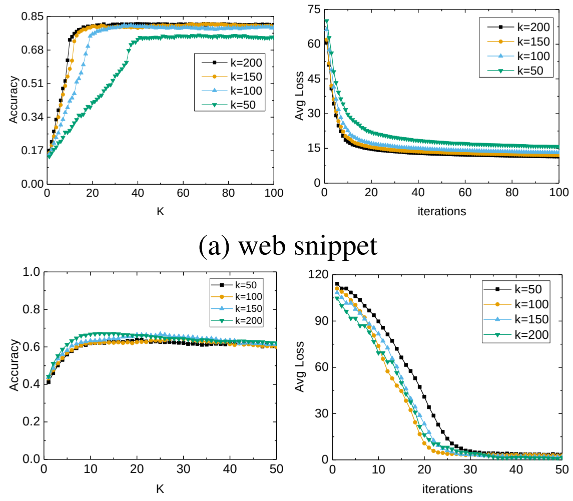
Our three models are implemented in Python using TensorFlow[8]. For both datasets, we use the pre-trained 300-dimensional word embeddings from Wikipedia by GloVe, and it is fixed during training. For each out-of-vocab word, we sample a random vector from a normal distribution. In practice, we use a regular learning rate 0.00001 for both dataset. We set the regularization factor $\lambda = 1, \alpha = 1, \lambda_1 = 0.6, \lambda_2 = 0.4$. In initialization, all weight matrices are randomly initialized with the uniformed distribution in the interval $[0, 0.001]$ for web snippet, and $[0, 0.0001]$ for 20Newsgroups.

4.2 Classification Accuracy

We perform text classification tasks on Web Snippet dataset and 20Newsgroups. For the Web Snippet, we follow its original partition that consists of 10060 training documents and 2280 test documents. On 20Newsgroups, we we keep 60% documents for training and 40% for testing as in (Zhu and Xing, 2011). We adopt the SVM as the classifier with the document representations learned by topic models. Figure 4 reports the convergence curves of loss and accuracy over iterations. The results show that the loss and accuracy of our method can achieve convergence after almost 100 epochs on web snippets and 50 epochs on 20Newsgroups with appropriate learning rate. Table 1 reports the classification accuracy on both datasets under different methods with different settings on the number of topics $K = \{50, 100, 150, 200, 250\}$. We can found that 1) The NSTCSG yields the highest accuracy, followed by NGSTC, NSTCE and NSTC which all outperform the DocNADE, GLDA, STC and LDA. 2) The embedding based models (NSTCSG, NGSTC, NSTCE, NSTC, DocNADE and GLDA) generate better document representations than STC and LDA separately, demonstrating the representative power of neural networks based on word embeddings. 3) Sparse models (NSTCSG, NGSTC, NSTCE, NSTC and STC) are superior to non-sparse models NTM and LDA separately. It indicates that sparse topic models are more suitable to short documents. 4) The NSTCSG perform better than NGSTC, followed by NSTC, which il-

[5] http://bigml.cs.tsinghua.edu.cn/ jun/stc.shtml/
[6] https://github.com/huashiyiqike/TMBP/tree/master/DocNADE
[7] https://github.com/rajarshd/Gaussian_LDA
[8] https://www.tensorflow.org/

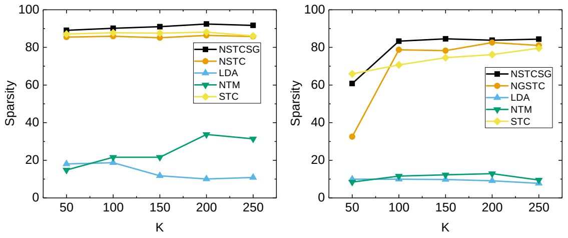
lustrates both sparse group lasso and group lasso penalty are contribute to learning the document representations with clear semantic explanations. 5) The NSTCE and NSTC have similar performances on two datasets.



(a) web snippet

(b) 20Newsgroup

Figure 4: The loss and accuracy curves with the iterations on two datasets,on different number of topic K settings.



(a) word codes

(b) document codes

Figure 5: The average sparsity ratio of word and document codes.

4.3 Sparse Ratio

We further compare the sparsity of the learned latent representations of words and documents from different models on Web Snippet.

Word code: For each word n, we compute the average word code and average sparsity ratio of them as in (Zhu and Xing, 2011). Figure 5a presents the average word sparse ratio of word codes discovered by different models for Web Snippet. Note that the NGSTC can not yield sparse word codes but sparse document codes. We can see that 1) The NSTCSG learns the sparsest word codes, followed by NSTC and STC, which perform much better than NTM and LDA. 2) The word codes discovered by LDA and NTM are very dense for lacking the mechanism to learn the focused topics. The sparsity in both models is mainly caused by the data scarcity. 3)The representations learned by sparse models (NSTCSG, NSTC and STC) are much sparser, which indicates each word concentrates on only a small number of topics in these models, and therefore the word codes are more clear and semantically concentrated. 4) Meanwhile, the sparse ratio of STC and NSTC are lower than NSTCSG. It proves the sparse group lasso penalty can easily allow to provide networks with a high level of sparsity.

Document code: We further quantitatively evaluate the average sparse ratio on latent representations of documents from different models, as shown in Figure 5b. We can see that 1) The NSTCSG yields the highest sparsity ratio, followed by NGSTC and STC, which outperform NTM and LDA by a large margin. 2) The document codes discovered by LDA and NTM are still very dense, while the representations learned by sparse models (NSTC and STC) are much sparser. It indicates the sparse models can discover focused topics and obtain discriminative representations of documents. 3) Similar to the word code, NGSTC outperforms NGSTC and STC. It demonstrates that the sparse group lasso penalty can achieve better sparsity than group lasso and lasso. 4) The sparsity ratios of sparse models increase with the topic numbers. The possible reason is that the sparse models tend to learn the focused topic number which approaches to the real topic number, and an increasing number of redundant topics can be discarded. 5) The NSTCSG inherits the advantages of NSTC and NGSTC, which can achieve the sparse topic representations of words and documents.

4.4 Quality of Extracted Topics and Representations

In table 2, we show the top-9 words of learned focused topics in 20 Newsgroups datasets. For each topic, we list top-9 words according to their probabilities under the corresponding topic. It is obvious that the learned topics are clear and meaningful. Such as *economics*, *hockey*, *games*, *play*, *ball* in the topic about sport. In Figure 6, we also use the 2-dimensional t-SNE method to get the visu-

Table 1: Classification accuracy of different models on Web snippet and 20NG, with different number of topic K settings.

Dataset	**Snippet**					**20NG**				
k	**50**	**100**	**150**	**200**	**250**	**50**	**100**	**150**	**200**	**250**
LDA	0.682	0.592	0.573	0.615	0.583	0.545	0.615	0.607	0.613	0.623
STC	0.678	0.699	0.724	0.731	0.723	0.602	0.631	0.647	0.652	0.654
DocNADE	0.656	0.656	0.645	0.646	0.647	0.682	0.670	0.646	0.583	0.573
GLDA	0.669	0.689	0.675	0.670	0.623	0.367	0.438	0.465	0.496	0.526
NSTC	0.734	0.756	0.791	0.793	0.789	0.634	0.671	0.682	0.690	0.72
NSTCE	0.739	0.778	0.801	0.803	0.810	0.631	0.681	0.682	0.701	0.721
NGSTC	0.773	0.792	0.813	0.811	0.821	0.670	0.681	0.701	0.712	0.737
NSTCSG	0.788	0.813	0.821	0.823	0.829	0.665	0.687	0.691	0.717	0.735

Table 2: Top Words of Learned Topics for 20Newsgroups.

computer	**sport**	**drug**	**weapon**	**space-flight**	**atheism**	**medication**	**politics**
computer	hockey	tobacco	nuclear	nasa	matthew	cancer	turkey
windows	games	drug	guns	flyers	state	insurance	south
ibm	motorcycl	fallacy	crime	space	atheism	technology	bill
drive	team	aids	booming	air	book	life	adress
disk	play	hiv	controller	statelite	god	hiv	congress
system	groups	dades	firearms	send	jesus	des	rockefeller
dos	came	illeg	military	launch	truth	patients	cosmo
key	rom	same	wiring	apartment	faq	water	american
hardware	ball	adict	neutral	la	church	health	slave

alization of the learned latent representations for Web Snippet and 20Newsgroups Dataset with 200 topics. For Web Snippet, we sample 10% of the whole dataset. For 20newsgroups, we sample 30% of the dataset. It is obvious to see that all documents are clustered into 8 and 20 distinct categories. It proves the semantic effectiveness of the documents codes learned by our model.

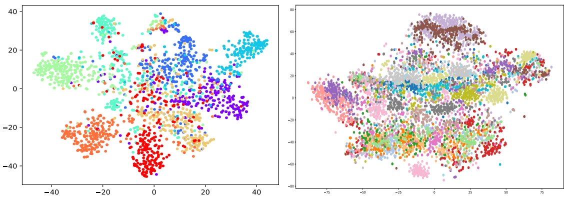


Figure 6: T-SNE embeddings of learned document representations for Web Snippet and 20NewsGroups. Different colors mean different categories.

5 Conclusion

In this paper, we propose a novel neural sparsity-enhanced topic model NSTC, which improves STC by incorporating the neural network and word embeddings. Compared with other word embedding based and neural network based topic models, it overcomes the computation complexity of topic models, and improve the generation of representation over short documents. We present three variants of NSTC to illustrate the great flexibility of our framework. Experimental results demonstrate the effectiveness and efficiency of our models. For future work, we are interested in various extensions, including combining STC with autoencoding variational Bayes (AVB).

References

Lu Bai, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. Group sparse topical coding: from code to topic. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, pages 315–324.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng

Ji. 2015. A novel neural topic model and its supervised extension. In *AAAI*.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian lda for topic models with word embeddings. In *ACL*.

Karol Gregor and Yann LeCun. 2010. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. pages 399–406.

Matthias Heiler and Christoph Schnörr. 2006. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *Journal of Machine Learning Research* 7(Jul):1385–1407.

Weihua Hu and Junichi Tsujii. 2016. A latent concept topic model for robust topic inference using word embeddings. In *The 54th Annual Meeting of the Association for Computational Linguistics*. page 380.

Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. 2010. Fast inference in sparse coding algorithms with applications to object recognition. *arXiv preprint arXiv:1010.3467* .

Hugo Larochelle and Stanislas Lauly. 2012a. A neural autoregressive topic model. In *NIPS*.

Hugo Larochelle and Stanislas Lauly. 2012b. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*. pages 2708–2716.

Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. 2014. The dual-sparse topic model: mining focused topics and focused terms in short text. In *WWW*.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*. pages 1614–1623.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics* 3:299–313.

Min Peng, Qianqian Xie, Jiajia Huang, Jiahui Zhu, Shuang Ouyang, Jimin Huang, and Gang Tian. 2016. Sparse topical coding with sparse groups. In *WAIM*.

Fei Tian, Bin Gao, Di He, and Tie-Yan Liu. 2016. Sentence level recurrent topic model: Letting topics speak for themselves. *CoRR* abs/1604.02038.

Sinead Williamson, Chong Wang, Katherine A. Heller, and David M. Blei. 2010. The ibp compound dirichlet process and its application to focused topic modeling. In *ICML*.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017. A correlated topic model using word embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence.[doi¿ 10.24963/ijcai. 2017/588]*.

Jun Zhu and Eric P. Xing. 2011. Sparse topical coding. *CoRR* abs/1202.3778.