Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

A Novel Neural Topic Model and Its Supervised Extension

Ziqiang Cao[1] **Sujian Li**[1] **Yang Liu**[1] **Wenjie Li**[2] **Heng Ji**[3]
[1]Key Laboratory of Computational Linguistics, Peking University, MOE, China
[2]Computing Department, Hong Kong Polytechnic University, Hong Kong
[3]Computer Science Department, Rensselaer Polytechnic Institute, USA
{ziqiangyeah, lisujian, pku7yang}@pku.edu.cn cswjli@comp.polyu.edu.hk jih@rpi.edu

Abstract

Topic modeling techniques have the benefits of modeling words and documents uniformly under a probabilistic framework. However, they also suffer from the limitations of sensitivity to initialization and unigram topic distribution, which can be remedied by deep learning techniques. To explore the combination of topic modeling and deep learning techniques, we first explain the standard topic model from the perspective of a neural network. Based on this, we propose a novel neural topic model (NTM) where the representation of words and documents are efficiently and naturally combined into a uniform framework. Extending from NTM, we can easily add a label layer and propose the supervised neural topic model (sNTM) to tackle supervised tasks. Experiments show that our models are competitive in both topic discovery and classification/regression tasks.

Introduction

The real-world tasks of text categorization and document retrieval rely critically on a good representation of words and documents. So far, state-of-the-art techniques including topic models (Blei, Ng, and Jordan 2003; Mcauliffe and Blei 2007; Wang, Blei, and Li 2009; Ramage et al. 2009) and neural networks (Bengio et al. 2003; Hinton and Salakhutdinov 2009; Larochelle and Lauly 2012) have shown remarkable success in exploring semantic representations of words and documents. Such models are usually embedded with latent variables or topics, which serve the role of capturing the efficient low-dimensional representation of words and documents.

Topic modeling techniques, such as Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003), have been widely used for inferring a low dimensional representation that captures the latent semantics of words and documents. Each topic is defined as a distribution over words and each document as a mixture distribution over topics. Thus, the semantic representations of both words and documents are combined into a unified framework which has a strict probabilistic explanation. However, topic models also suffer from certain limitations as follows. First, LDA-based models require prior distributions which are always difficult to define. Second, previous models rarely adopt n-grams beyond unigrams in document modeling due to the sparseness problem, though n-grams are important to express text. Last, when there is extra labeling information associated with documents, topic models have to do some task-specific transformation in order to make use of it (Mcauliffe and Blei 2007; Wang, Blei, and Li 2009; Ramage et al. 2009), which may be computationally costly.

Recently, deep learning techniques also make low dimensional representations (i.e., distributed representations) of words (i.e., word embeddings) and documents (Bengio et al. 2003; Mnih and Hinton 2007; Collobert and Weston 2008; Mikolov et al. 2013; Ranzato and Szummer 2008; Hinton and Salakhutdinov 2009; Larochelle and Lauly 2012; Srivastava, Salakhutdinov, and Hinton 2013) feasible. Word embeddings provide a way of representing phrases (Mikolov et al. 2013) and are easy to embed with supervised tasks (Collobert et al. 2011). With layer-wise pre-training (Bengio et al. 2007), neural networks are built to automatically initialize their weight values. Yet, the main problem of deep learning is that it is hard to give each dimension of the generated distributed representations a reasonable interpretation.

Based on the analysis above, we can see that current topic modeling and deep learning techniques both exhibit their strengths and defects in representing words and documents. A question comes to our mind: Can these two kinds of techniques be combined to represent words and documents simultaneously? This combination can on the one hand overcome the computation complexity of topic models and on the other hand provide a reasonable probabilistic explanation of the hidden variables.

In our preliminary study we explain topic models from the perspective of a neural network, starting from the fact that the conditional probability of a word given a document can be seen as the product of the probability of a word given a topic (word-topic representation) and the probability of a topic given the document (topic-document representation). At the same time, to solve the unigram topic distribution problem of a standard topic model, we make use of the word embeddings available (Mikolov et al. 2013) to represent n-grams. Based on the neural network explanation and n-gram representation, we propose a novel neural topic model (NTM) where two hidden layers are constructed to efficiently acquire the n-gram topic and topic-document rep-

resentations as in a topic model. Then, the product of these two layers indicates how likely an n-gram appears in a document.

The most relevant work to ours is the model proposed by (Keller and Bengio 2005), which adopts the multi-layer perceptrons to capture the distributed representations of both words and documents, but not all the layers are interpretable. Different from (Keller and Bengio 2005), our model follows the probabilistic characteristics of topic models. Thus, the distributed representations of words and documents have a reasonable probabilistic explanation. Furthermore, under the framework of a neural network, our model can be naturally and easily extended to supervised tasks such as document categorization and rating, through adding a label layer.

Finally, our proposed models are verified on three typical document learning tasks, namely multi-class classification, multi-label classification and rating regression. Results show the topic representations of our models have been improved greatly compared to state-of-the-art approaches.

Related Work

Topic Model

A topic model is normally designed for statistically discovering the abstract "topics" hidden in a collection of documents. One of the early topic models is probabilistic latent semantic indexing (pLSI) (Hofmann 1999). pLSI models each word in a document as a sample from a mixture model, where topics are represented as the multinomial random variables and documents as a mixture of topics. Latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003), the most widely used topic model, can be seen as a generalization of pLSI. In LDA, Dirichlet conjugate priors are introduced for both the word multinomial distributions over a topic and topic multinomial distributions over a document.

How to extend topic models to different supervised problems is not well-studied. Usually the design is case by case. Some of state-of-the-art supervised topic models are given below. To adapt to regression tasks, (Mcauliffe and Blei 2007) proposed the sLDA model (named sLDAr in this paper) by adding to LDA a response variable associated with each document and then defining for the variable a Gaussian distribution whose mean value was computed by a linear regression of topics. Based on their work, (Wang, Blei, and Li 2009) proposed the multi-class sLDA model (named sLDAc in this paper) by adding a softmax classifier for a standard LDA and used it in the multi-class classification (MCC) task. Also for this task, the DiscLDA model proposed by (Lacoste-Julien, Sha, and Jordan 2009) introduced a linear transformation matrix to project the document-topic distribution to a class-related space. (Ramage et al. 2009) designed the labeled LDA model (L-LDA) by defining a one-to-one correspondence between the latent topics and the document labels, and applied this model to the multi-label classification (MLC) task. (Zhu, Ahmed, and Xing 2009) utilized the max-margin principle and designed the MedLDA models for regression and classification respectively. It is noted that DiscLDA and MedLDA are discriminative while sLDAr, sLDAc and L-LDA are generative.

Deep Learning in NLP

Deep learning (DL) techniques are capable of automatically learning low dimensional representations of things. Since the creative work of the neural network language model (NNLM) (Bengio et al. 2003), DL has become more and more popular in the field of natural language processing (NLP). A great deal of research (Mnih and Hinton 2007; Collobert and Weston 2008; Mikolov et al. 2013) has used various DL methods to explore the generation of word embeddings. At the same time, some researchers tried to model documents with layer-wise deep learning tools, including auto-encoders (Ranzato and Szummer 2008), restricted Boltzmann machine (RBM) (Hinton and Salakhutdinov 2009), document neural autoregressive distribution estimators (DocNADE) (Larochelle and Lauly 2012) and deep Boltzmann machine (DBM) (Srivastava, Salakhutdinov, and Hinton 2013). These methods use the word count vector of a document as input, and then related DL tools to build hidden layers which can synthesize the input. Like topic models, the hidden layers in the deep networks provide the low-dimensional representation of documents. There also exists some research which attempted to capture jointly the distributed representations of words and documents mainly from the perspective of web search or document classification (Huang et al. 2013; Keller and Bengio 2005). (Huang et al. 2013) aimed to represent words and documents with low-dimensional vectors through factorizing words or documents into letter n-grams. (Keller and Bengio 2005) assumed word representation and document representation are linked and built a model named Neural Network for Text Representation (NNTR).

Neural Topic Model

In this section, we first explain topic models from the view of neural networks. Based on this explanation, we propose our neural topic model (NTM) and its extension (sNTM).

Neural Network View of Topic Models

Generally, for topic models, each document is represented as a mixture of topics, where each topic defines a probability distribution over words. In other words, documents in a corpus share the same set of K topics, and each document owns a different mixture of topics. To formulate, suppose d is a document and w is a word in d. Topic models compute the conditional probability $p(w|d)$ as the combination of word-topic distribution and topic-document distribution:

$$p(w|d) = \sum_{i=1}^{K} p(w|t_i)p(t_i|d), \tag{1}$$

where t_i is a latent topic and K is the pre-defined topic number. Let $\phi(w) = [p(w|t_1), \cdots, p(w|t_K)]$ and $\theta(d) = [p(t_1|d), \cdots, p(t_K|d)]$, where ϕ is shared among the corpus and θ is document-specific. Then Eq. 1 can be represented as the vector form:

$$p(w|d) = \phi(w) \times \theta^T(d). \tag{2}$$

Based on Eq. 2, we can explain topic models from the view of a neural network, where $\phi(w)$ functions as the look-up layer for words with the sigmoid activation function, and

$\theta(d)$ as a look-up layer for documents with the softmax activation function. The output of the neural network is calculated as the dot product of $\phi(w)$ and $\theta(d)$. To summarize, the LDA model is converted to a neural network with input word w and document d, and output the conditional probability $p(w|d)$.

As stated in the first section, topic models such as LDA provide a probabilistic explanation of topics, but suffers from imprecise prior knowledge and restricted unigram-topic representation. In our work, extending from the neural network of topic models, we improve the generation of word-topic distributions and topic-document distributions with deep learning techniques, alleviating the problems of inference and prior knowledge.

Neural Topic Model (NTM)

Based on the neural network perspective of topic models, we propose our neural topic model (NTM) as illustrated in the left frame of Figure 1. The right part of NTM is designed to generate semantic representations of documents, while the left part aims to capture word n-grams. With documents and n-grams as the input to the neural network, two hidden layers, namely the n-gram-topic layer and the topic-document layer, are designed to generate the n-gram topic and topic-document distributions, similar to ϕ and θ in topic models. The dot symbol on the top denotes a dot product computation, outputting a probabilistic score, like the conditional probability $p(w|d)$ in topic models.

As we know, the direct introduction of n-grams into topic models will cause the sparseness problem. To allow our method to model beyond word unigrams, we make use of the word embeddings trained by a large corpus of free text and sum them up to represent n-grams. The main reasons of such implementation are as follows. First, according to the work of (Mikolov et al. 2013), n-grams can be roughly represented by the accumulation of the n word embeddings. Second, the generated word embeddings have been verified to well capture the semantics of words (Huang et al. 2012). Then, we get an n-gram embedding layer lying between the n-grams and the word-topic layer.

To be explicit, we formally explain each layer of NTM below.

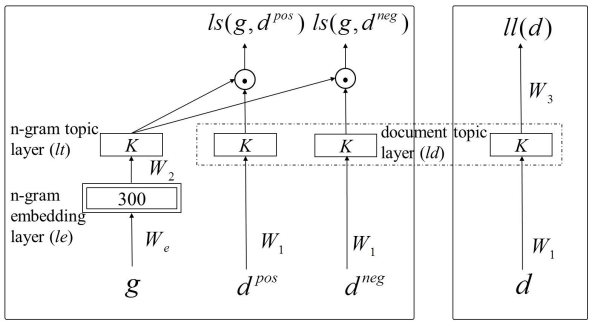

Figure 1: Unsupervised Neural Topic Model (NTM) and Its Extension (sNTM)

Input layer (g, d): An n-gram $g = w_1, \cdots, w_n$ and a document ID $d \in D$, where D is a document set.

n-gram embedding layer ($le \in \mathbb{R}^{1\times 300}$): The goal of this layer is to represent each n-gram with a distributed embedding representation. Here, we use the available tool word2vec[1] trained on a large Google News dataset (about 100 billion words), which provides an embedding matrix W_e for a vocabulary of about 3 million words or phrases. Each word or phrase is represented by a 300-dimensional embedding vector. Then for any n-gram g, its embedding representation is obtained as follows: if g is in the vocabulary, we directly use its embedding representation. Otherwise, g is represented by summing up its n word embeddings. It seems more reasonable to update W_e in the training step. However, considering our dataset is much smaller than the one used for word2vec, we fix this layer, which also makes our model independent on the vocabulary size. We use double planes in Figure 1 to indicate this fact.

n-gram-topic layer ($lt \in \mathbb{R}^{1\times K}$): This layer stands for the topic representation of the input n-grams. Supposing the topic number is K, each word is denoted as a K-dimensional vector. This vector follows a probabilistic form, similar to ϕ in topic models, by applying the sigmoid function.

$$lt(g) = sigmoid(le(g) \times W_2) \tag{3}$$

where $W_2 \in \mathbb{R}^{300\times K}$ denotes the weight matrix between the n-gram embedding layer and the n-gram-topic layer. To conform to the framework of a topic model, a topic in our model is also seen as a multinomial distribution over n-grams. Thus, we normalize the n-gram topic distribution and use $\widetilde{lt}(g)$ to denote the normalized form.

$$\widetilde{lt_i}(g) = \frac{1}{z_i} lt_i(g) = \frac{1}{\sum_g lt_i(g)} lt_i(g) \tag{4}$$

where $z_i = \sum_g lt_i(g)$ is the normalized factor. The computation of z_i is proportional to the number of n-grams and time-consuming. Since a topic is composed of an infinite number of n-grams although we deal with only a small part of them, it is intractable to compute the value of z_i. Here, we make a simplified assumption: each z_i equals to the same constant Z. Under this assumption, we use the unnormalized value $lt(g)$ in Eq. 3 to simulate the word topic distribution.

Topic-document layer ($ld \in \mathbb{R}^{1\times K}$): With a document look-up matrix table $W_1 \in \mathbb{R}^{|D|\times K}$, we can convert a document d to a vector representation $W_1(d,:)$. This layer aims at converting $W_1(d,:)$ to a topic distribution, similar to θ in topic models. Here, we adopt a softmax function to keep the probabilistic constraint.

$$ld(d) = softmax(W_1(d,:)) \tag{5}$$

Scoring layer ($ls \in \mathbb{R}$): This layer outputs the matching score of an n-gram g and a document d, with the dot product of $lt(g)$ and $ld(d)$. The outputted score $ls(g,d)$ is a probabilistic value between 0 and 1, similar to the conditional probability of $p(g|d)$.

$$ls(g,d) = lt(g) \times ld(d)^T \tag{6}$$

[1] https://code.google.com/p/word2vec/

Supervised Neural Topic Model (sNTM)

Some research has transformed topic models to adapt to supervised tasks, but they indeed face the problems of modeling extra parameters for the labels and defining prior distributions. Though NTM can be seen as an extension of topic models, it is in nature a neural network. Therefore, it is relatively flexible and easy to convert into a supervised model by adding a new layer for the labeling information.

The right frame of Figure 1 shows how we extend NTM to the supervised tasks of classification or regression. In the supervised tasks, a good representation of documents is vital to the final results. Fortunately, the topic-document layer provides a compact and semantic representation of documents. Thus, in sNTM, a label layer (ll) is designed on the top of the topic-document layer, parallel to the scoring layer. The topic-document layer serves the input to the label layer. In turn, with the labeling results, sNTM can tune the topic distributions of documents, further improving the generation of word topic distributions as well as the scoring results for the (n-gram, document) pairs. Here, the labeling results ($ll(d)$) can be computed as:

$$ll(d) = f(ld(d) \times W_3) \tag{7}$$

where the matrix W_3 denotes the weights of each topic contributing to the labeling results. $f(\cdot)$ indicates an activation function which depends on the property of the label.

Training

This section details the training procedure for our models. NTM is designed to take advantage of both the huge amount of unlabeled documents and n-grams to explore their semantic representation. Given a (n-gram, document) pair (g, d), its score is denoted by $ls(g, d)$, similar to the probability of $p(g|d)$. It is ideal to directly estimate such a conditional probability to supervise the training process. However, it is difficult to get a precise estimation for these conditional probabilities, especially computed according to one specific document. Here, we follow a pairwise ranking approach introduced by (Collobert et al. 2011). The basic idea is, a higher score is given if the n-gram g is contained in the document d, and a lower score otherwise. Specifically, assuming g is an n-gram and d^{pos} is a document containing g, we randomly sample another document d^{neg} which does not contain g. For the positive example (g, d^{pos}) and negative sample (g, d^{neg}), we keep their scores a margin Ω with the experience value of 0.5. To achieve this, the following cost function is minimized:

$$c(g, d^{pos}, d^{neg}) = \max(0, \Omega - ls(g, d^{pos}) + ls(g, d^{neg})) \tag{8}$$

According to the cost function and the available document labels, the training algorithm for NTM (sNTM) is shown in Algorithm 1. In NTM, two matrices, i.e., the document look-up table W_1 and the embedding-topic weight matrix W_2 , need to be trained based on the cost function as in Eq. 8 (Lines 3-6). We use the back propagation (BP) algorithm to adjust their weights. Here, stochastic gradient descent (SGD) with L_2 norm regularization is adopted. The learning rate is set to 0.01 and the regularization factor is set to 0.001. For sNTM, an extra weight matrix W_3 is updated based on the labels of the documents using the BP algorithm, which also provides an adjustment for W_1 (Lines 7-10). For testing, we need to perform an inference step to re-compute a proper W_1 for new documents. This is similar to the training process of NTM while the remaining parameters are fixed.

Algorithm 1 Training Algorithm for NTM(sNTM)

Input: $T = \{(g, d^{pos})\}$ where $d^{pos} \in D$ is a document containing the n-gram g; For sNTM, $\{lable(d)\}$ where $label(d)$ is the correct label of d.

1: Pre-train;
2: **repeat**
3: FOREACH $(g, d^{pos}) \in T$:
4: Sample a document d^{neg} where $(g, d^{neg}) \notin T$;
5: IF $c(g, d^{pos}, d^{neg}) > 0$:
6: Update W_1, W_2 using BP;
7: IF sNTM Then:
8: FOREACH $d \in D$:
9: Compute label error;
10: Update W_1, W_3 using BP;
11: **until** convergence

Due to the deep structure in our model, BP is prone to be trapped in local optima. Hence, we design a pre-training procedure for better initializing W_1 and W_2 (Line 1 in Algorithm 1). First, we use an auto-encoder between the n-gram embedding layer and the n-gram topic layer. Given the n-gram embedding layer $le(g)$, we seek the proper values of W_2 to reconstruct it. Specifically, with the constraint that the weights between the two layers are symmetric, we can get:

$$le'(g) = \text{sigmoid}(le(g) \times W_2) \times W_2^T \tag{9}$$

where $le'(g)$ is the reconstructed embedding layer. Considering the optimization goal that le' is as close to le as possible, we can tune the values of W_2. Next, to initialize W_1, we conduct a simplified implementation based on W_2. The topic representation of a document is obtained by summing up and normalizing the n-gram topic representation of all the n-grams included in the document.

Experimental Results

Data and Setting

To evaluate both NTM and its supervised extension sNTM, we purposely collect the text corpora which include the labeling information. Here, three datasets, namely 20 Newsgroups[2], Wiki10+ (Zubiaga 2012) and Movie review data (Pang and Lee 2005), are used. 20 Newsgroups has a collection of about 20,000 documents organized into 20 different classes. Wiki10+ dataset is composed of Wikipedia articles, each of which is associated with one or more social tags retrieved from delicious.com. From Wiki10+ we find the most frequent 25 social tags and only keep those documents containing any of these tags. Then we have a collection of 17,325 documents with 2.6 tags per document on average. Movie review data (MRD) is a collection of movie review documents with sentiment scaling scores, and the rating of movie reviews can be seen as a regression problem.

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

Two sets of experiments are designed: First, we select 20 Newsgroups to examine the generated topics of the NTM model and compare them with two state-of-the-art unsupervised topic models LDA (Blei, Ng, and Jordan 2003) and DocNADE (Larochelle and Lauly 2012). DocNADE is reported to outperform the Replicated Softmax model (Hinton and Salakhutdinov 2009), which is thus not used as a baseline here. Second, we evaluate the performance of our topic models on document-oriented supervised tasks including multi-class classification (MCC), multi-label classification (MLC), and regression. As for the function $f(\cdot)$ in sNTM, we adopt the $softmax$ function for the MLC task and $sigmoid$ functions for both the MLC and regression tasks. One baseline named *Word2Vec*, simply sums up the word embeddings of those words in a document to represent a document. Three unsupervised models, NTM, LDA, and DocNADE, are also used as baselines. These four baselines can not be applied directly to the supervised tasks, and thus we apply the $softmax$ or $sigmoid$ function on their document representation for producing the final labels. At the same time, three supervised topic models sLDAc (Wang, Blei, and Li 2009), L-LDA (Ramage et al. 2009) and sLDAr (Mcauliffe and Blei 2007), which have been briefed in Section 2, are selected as the baselines respectively for the MCC, MLC and regression tasks. The datasets of 20 Newsgroups, Wiki10+ and Movie review data are respectively used as the corpus for accomplishing the tasks of MCC, MLC and regression. Table 1 describes the experimental data, including the size of the training part and test part, the average length in words per document, and task.

Dataset	20 Newsgroups	Wiki10+	Sentiment Scale
Train.Size	11149	11550	3337
Test.Size	7403	5775	1669
Avg. Length	135	1704	176
Task	MCC	MLC	Regression

Table 1: Experimental Datasets

Qualitative Inspection of Topics

Compared with other topic models, the advantage of our models is that they can deal with n-grams, instead of single words. In the practical implementation, we only consider unigrams and bigrams, since an n-gram tends to be meaningless and its embedding representation becomes less precise as n increases. A unigram can also be seen as a special case of a bigram where one word is null. Here, we inspect the topics by their dominated bigrams (words) qualitatively.

We manually examine all the topics generated by LDA, DocNADE, NTM, sNTM and sLDAc using the 20 Newsgroups corpus. The topic numbers of these models are all set to 100. From each model we pick out three sample topics which are closely related to the three classes, namely Class 1 (alt.atheism), Class 16 (soc.religion.christian), Class 20 (talk.religion.misc). Table 2 and Table 3 show the typical topic words (or bigrams) which have the strongest positive connections with each topic. By looking into the topics, we find that some typical topic words in LDA are numbers or words which contain one or two characters. DocNADE performs better than LDA and most typical topic words are meaningful. We also find that some typical topic bigrams generated by NTM, such as "sacred scripture" and "religious fanaticism", are more appropriate to expressing a topic. On contrary, LDA and DocNADE can only use single words to represent topics, which often distort their real meaning under the bag-of-words assumption. For example, the topic bigram "video decoder" generated by NTM reflects the topic about software topic, but LDA improperly catches the word "video" in the software topic.

As we qualitatively compare the NTM topics in Table 2 with the sNTM topics in Table 3, we find that NTM contains more unigrams to represent topics while sNTM selects more dominate bigrams in its topics. For example, for Class 20, the sNTM topic contains a series of specific church names comprised of two words, while the NTM topic contains some single words. For Class 1, the sNTM topic lists the names of two philosophers while the NTM topic even contains some general words such as "science" and "proof". This reveals that the labeling class information in sNTM is helpful to supervise the generation of the topic distributions. As for sLDAc, we find that there are always some overlaps between two topics, because only single words are allowed to express the topics. For example,"christian" occurs simultaneously in Topic 38 and 16, and "religion" occurs in Topic 4 and 16.

Evaluation on Supervised Tasks

In this subsection, we focus on analyzing the performance of our topic models on the MCC task. Next we exhibit that our models can also perform well on the MLC and regression tasks. Normally, the better the performance is, the more reasonable document representation the model generates.

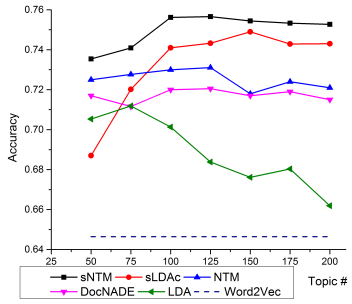

Figure 2: MCC Performance

Evaluation on the MCC Task To evaluate the MCC performance, we adopt the *Averaged Accuracy* over all classes. Figure 2 shows the *accuracy* of different models with respect to number of topics. Generally, we can see that the performance of the models gets better as the number of topics increases and achieves their best when the number of topics is set around 100. Overall, supervised methods (sNTM and sLDAc) are superior to unsupervised ones (Word2Vec,

Class 1(alt.atheism)			Class 16(soc.religion.christian)			Class 20(talk.religion.misc)		
LDA(4)	DocNADE(31)	NTM(47)	LDA(66)	DocNADE(27)	NTM(59)	LDA(5)	DocNADE(8)	NTM(72)
belief	moral	atheism	jesus	jesus	god	lds	church	lds church
god	atheism	creation	god	god	jesus	istanbul	catholic	pope
truth	keith	human being	christ	sin	catholic faith	georgia	orthodox	protestant
reason	religion	true nature	s	lord	christ	ermeni	holy	nestorius
atheist	system	science	lord	heaven	saint	york	tradition	religious belief
christian	islam	sacred scripture	sin	church	bible	church	doctrine	catholic church
bible	belief	theism	bible	life	homosexuality	ankara	movement	religious fanaticism
religion	muslim	proof	heaven	hell	heaven	##	spirit	theology

Table 2: Example Topics Generated by LDA, DocNADE and NTM. (**The digits denote the specific class number or topic numbers.**)

Class 1(alt.atheism)		Class 16(soc.religion.christian)		Class 20(talk.religion.misc)	
sLDAc(4)	sNTM(33)	sLDAc(38)	sNTM(29)	sLDAc(16)	sNTM(52)
atheism	atheist	christian	jesus	christian	catholic church
moral	historical revisionist	god	god	lds	episcopal church
keith	secular humanism	truth	heaven	bible	protestant church
religion	blasphemous	bible	catholic liturgy	god	lds church
system	ludwig wittgenstein	belief	mormon doctrine	religion	apostolic church
islam	friedrich nietzsche	church	christ	church	christian church
belief	biblical interpretation	jesus	bible	objective	mormon church
muslim	science	rutgers	homosexuality	morality	lutheran church

Table 3: Example Topics Generated by sLDAc and sNTM

NTM, DocNADE and LDA). sNTM achieves significant improvement [3] against sLDAc, and NTM shows significant superiority against DocNADE, LDA and Word2Vec. In addition, we can see that sNTM performs much more stably than sLDAc. In fact, with a class variable defined in the generation process, the inference time of sLDAc is 10 times longer than sNTM. Since the average length of a document in 20 Newsgroups is short and there are many general words in it, a document representation with a simple accumulation of word embeddings always distorts the real meaning of the document. For this reason, Word2Vec performs the worst.

As we know, the only difference between NTM and sNTM is that there is an extra label layer in sNTM. With the label layer added, sNTM can further tune the topic distributions generated by NTM and promote the classification performance. The previous subsection has compared and analyzed the generated topics, which can explain why sNTM performs better than NTM on the MCC task.

Evaluation on the MLC and Regression Tasks To further verify that sNTM can flexibly adapt to various supervised tasks, we further test it on the social tagging (MLC) task and movie rating prediction (regression) task. Here we only compare the best performance of our neural topic models with that of the other models.

To evaluate the performance of MLC, the left three columns of Table 4 score each model using the Micro-F_1 and Macro-F_1 measures (Lewis et al. 2004). The former allows larger classes to dominate the performance while the latter assigns an equal weight to all classes. To evaluate the regression task, we adopt the predictive R^2 (pR^2) metric (Mcauliffe and Blei 2007), which measures how well a regression model predicts responses for new observations. The right two columns of Table 4 illustrate the pR^2 scores on different models. From the results, we can draw the same conclusions as in the MCC task: supervised topic models exhibit explicit advantage over unsupervised topic models. This further verifies that labeled information can supervise the generation of topics. In the MLC task, the Macro-F_1 and Micro-F_1 scores of sNTM are stably better than L-LDA. We analyze the results and find that the main contribution comes from the bigram topic words generated by sNTM. Here, Word2Vec exhibits an acceptable performance, even better than LDA. This is because the average length of each document in Wiki10+ is relatively long and some keyphrases always occur repeatedly in a document. In such situation, an accumulation of word embeddings is capable of representing a document to some extent.

In the regression task, sNTM achieves significant improvement against sLDAr. We observe that the topic bigrams such as "not good" and "at best" generated by sNTM are helpful to rating the movie reviews. For example, sNTM takes the bigrams "at best", "at least", "worst" and "waste time" in the same topic while sLDAr takes the words "best" "well" "good" and "first" in one topic. We can also see that Word2Vec performs terribly worst, because the accumulation of word embeddings neutralizes the meaning of those negation words and exclamation words in the documents.

Conclusions

In this paper, we propose a novel neural topic model **NTM** which combines the advantages of both topic models and neural networks. In NTM, deep learning techniques are used to conduct inference under the framework of a topic model. Compared with the standard topic models, NTM overcomes

[3] t-test with $p \leq 0.05$

MLC	Macro-F_1	Micro-F_1	Regression	pR^2
sNTM	0.6508	0.6627	sNTM	0.4666
L-LDA	0.6280	0.6451	sLDAr	0.4250
NTM	0.4512	0.4805	NTM	0.2644
DocNADE	0.4451	0.4782	DocNADE	0.2347
LDA	0.2641	0.3325	LDA	0.1715
Word2Vec	0.3525	0.3633	Word2Vec	-3.6865

Table 4: Performance on the MLC and Regression Tasks.

the problem that only unigrams are allowed in the topic representation. Since NTM is in essence a neural network, we flexibly transform it to supervised tasks by adding a label layer and thus propose its supervised extension (sNTM). Experiments exhibit the high quality of the topics generated by our neural topic models and competitive performance on supervised tasks compared to state-of-the-art approaches.

Acknowledgments

We thank all the anonymous reviewers for their detailed and insightful comments on this paper. This work was partially supported by National High Technology Research and Development Program of China (No. 2012AA011101), National Key Basic Research Program of China (No. 2014CB340504), National Natural Science Foundation of China (No. 61273278 and No. 61272291), and National Key Technology R&D Program (No: 2011BAH10B04-03). The correspondence author of this paper is Sujian Li.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3:1137–1155.

Bengio, Y.; Lamblin, P.; Popovici, D.; and Larochelle, H. 2007. Greedy layer-wise training of deep networks. *Advances in neural information processing systems* 19:153.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

Hinton, G. E., and Salakhutdinov, R. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, 1607–1614.

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

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 873–882.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2333–2338.

Keller, M., and Bengio, S. 2005. A neural network for text representation. In *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*. 667–672.

Lacoste-Julien, S.; Sha, F.; and Jordan, M. I. 2009. Disclda: Discriminative learning for dimensionality reduction and classification. In *Advances in neural information processing systems*, 897–904.

Larochelle, H., and Lauly, S. 2012. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, 2717–2725.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research* 5:361–397.

Mcauliffe, J. D., and Blei, D. M. 2007. Supervised topic models. In *Advances in Neural Information Processing Systems*, 121–128.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mnih, A., and Hinton, G. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, 641–648.

Pang, B., and Lee, L. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 115–124.

Ramage, D.; Hall, D.; Nallapati, R.; and Manning, C. D. 2009. 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*, 248–256.

Ranzato, M., and Szummer, M. 2008. Semi-supervised learning of compact document representations with deep networks. In *Proceedings of the 25th international conference on Machine learning*, 792–799.

Srivastava, N.; Salakhutdinov, R. R.; and Hinton, G. E. 2013. Modeling documents with deep boltzmann machines. *arXiv preprint arXiv:1309.6865*.

Wang, C.; Blei, D.; and Li, F.-F. 2009. Simultaneous image classification and annotation. In *IEEE Conference on Computer Vision and Pattern Recognition 2009.*, 1903–1910.

Zhu, J.; Ahmed, A.; and Xing, E. P. 2009. Medlda: maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*.

Zubiaga, A. 2012. Enhancing navigation on wikipedia with social tags. *arXiv preprint arXiv:1202.5469*.