Topic Compositional Neural Language Model

Wenlin Wang[1] **Zhe Gan**[1] **Wenqi Wang**[3] **Dinghan Shen**[1] **Jiaji Huang**[2]
Wei Ping[2] **Sanjeev Satheesh**[2] **Lawrence Carin**[1]
[1]Duke University [2]Baidu Silicon Valley AI Lab [3]Purdue University

Abstract

We propose a Topic Compositional Neural Language Model (TCNLM), a novel method designed to simultaneously capture both the *global* semantic meaning and the *local* word-ordering structure in a document. The TCNLM learns the global semantic coherence of a document via a neural topic model, and the probability of each learned latent topic is further used to build a Mixture-of-Experts (MoE) language model, where each expert (corresponding to one topic) is a recurrent neural network (RNN) that accounts for learning the local structure of a word sequence. In order to train the MoE model efficiently, a matrix factorization method is applied, by extending each weight matrix of the RNN to be an ensemble of topic-dependent weight matrices. The degree to which each member of the ensemble is used is tied to the document-dependent probability of the corresponding topics. Experimental results on several corpora show that the proposed approach outperforms both a pure RNN-based model and other topic-guided language models. Further, our model yields sensible topics, and also has the capacity to generate meaningful sentences conditioned on given topics.

1 Introduction

A language model is a fundamental component to natural language processing (NLP). It plays a key role in many traditional NLP tasks, ranging from speech recognition (Mikolov et al., 2010; Arisoy et al., 2012; Sriram et al., 2017), machine translation (Schwenk et al., 2012; Vaswani et al., 2013) to image captioning (Mao et al., 2014; Devlin et al., 2015). Training a good language model often improves the underlying metrics of these applications, *e.g.,* word error rates for speech recognition and BLEU scores (Papineni et al., 2002) for machine translation. Hence, learning a powerful language model has become a central task in NLP. Typically, the primary goal of a language model is to predict distributions over words, which has to encode both the semantic knowledge and grammatical structure in the documents. RNN-based neural language models have yielded state-of-the-art performance (Jozefowicz et al., 2016; Shazeer et al., 2017). However, they are typically applied only at the sentence level, without access to the broad document context. Such models may consequently fail to capture long-term dependencies of a document (Dieng et al., 2016).

Fortunately, such broader context information is of a semantic nature, and can be captured by a topic model. Topic models have been studied for decades and have become a powerful tool for extracting high-level semantic structure of document collections, by inferring latent topics. The classical Latent Dirichlet Allocation (LDA) method (Blei et al., 2003) and its variants, including recent work on neural topic models (Wan et al., 2012; Cao et al., 2015; Miao et al., 2017), have been useful for a plethora of applications in NLP.

Although language models that leverage topics have shown promise, they also have several limitations. For example, some of the existing methods use only pre-trained topic models (Mikolov and Zweig, 2012), without considering the word-sequence prediction task of interest. Another key limitation of the existing methods lies in the integration of the learned topics into the language model; *e.g.*, either through concatenating the topic vector as an additional feature of RNNs (Mikolov and Zweig, 2012; Lau et al., 2017), or re-scoring the predicted distribution over words using the topic vector (Dieng et al., 2016). The former requires a balance between the number of RNN hidden units and

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

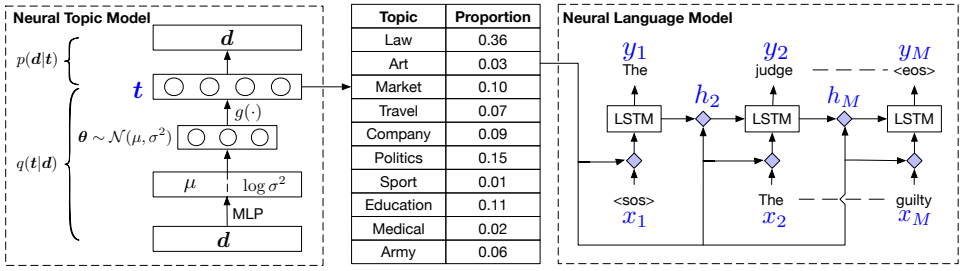

Figure 1: The overall architecture of the proposed model.

the number of topics, while the latter has to carefully design the vocabulary of the topic model.

Motivated by the aforementioned goals and limitations of existing approaches, we propose the Topic Compositional Neural Language Model (TCNLM), a new approach to simultaneously learn a neural topic model and a neural language model. As depicted in Figure 1, TCNLM learns the latent topics within a variational autoencoder (Kingma and Welling, 2013) framework, and the designed latent code $\boldsymbol{t}$ quantifies the probability of topic usage within a document. Latent code $\boldsymbol{t}$ is further used in a Mixture-of-Experts model (Hu et al., 1997), where each latent topic has a corresponding language model (expert). A combination of these "experts," weighted by the topic-usage probabilities, results in our prediction for the sentences. A matrix factorization approach is further utilized to reduce computational cost as well as prevent overfitting. The entire model is trained end-to-end by maximizing the variational lower bound. Through a comprehensive set of experiments, we demonstrate that the proposed model is able to significantly reduce the perplexity of a language model and effectively assemble the meaning of topics to generate meaningful sentences. Both quantitative and qualitative comparisons are provided to verify the superiority of our model.

2 Preliminaries

We briefly review RNN-based language models and traditional probabilistic topic models.

Language Model A language model aims to learn a probability distribution over a sequence of words in a pre-defined vocabulary. We denote $\mathcal{V}$ as the vocabulary set and $\{y_1, ..., y_M\}$ to be a sequence of words, with each $y_m \in \mathcal{V}$. A language model defines the likelihood of the sequence through a joint probability distribution

$$p(y_1, ..., y_M) = p(y_1) \prod_{m=2}^{M} p(y_m|y_{1:m-1}) \,. \tag{1}$$

RNN-based language models define the conditional probabiltiy of each word y_m given all the previous words $y_{1:m-1}$ through the hidden state $\boldsymbol{h}_m$:

$$p(y_m|y_{1:m-1}) = p(y_m|\boldsymbol{h}_m) \tag{2}$$

$$\boldsymbol{h}_m = f(\boldsymbol{h}_{m-1}, x_m) \,. \tag{3}$$

The function $f(\cdot)$ is typically implemented as a basic RNN cell, a Long Short-Term Memory (LSTM) cell (Hochreiter and Schmidhuber, 1997), or a Gated Recurrent Unit (GRU) cell (Cho et al., 2014). The input and output words are related via the relation $x_m = y_{m-1}$.

Topic Model A topic model is a probabilistic graphical representation for uncovering the underlying semantic structure of a document collection. Latent Dirichlet Allocation (LDA) (Blei et al., 2003), for example, provides a robust and scalable approach for document modeling, by introducing latent variables for each token, indicating its topic assignment. Specifically, let $\boldsymbol{t}$ denote the topic proportion for document d, and z_n represent the topic assignment for word w_n. The Dirichlet distribution is employed as the prior of $\boldsymbol{t}$. The generative process of LDA may be summarized as:

$$\boldsymbol{t} \sim \text{Dir}(\alpha_0), z_n \sim \text{Discrete}(\boldsymbol{t}) \,, w_n \sim \text{Discrete}(\boldsymbol{\beta}_{z_n}) \,,$$

where $\boldsymbol{\beta}_{z_n}$ represents the distribution over words for topic z_n, α_0 is the hyper-parameter of the Dirichlet prior, $n \in [1, N_d]$, and N_d is the number of words in document d. The marginal likelihood for document d can be expressed as

$$p(d|\alpha_0, \boldsymbol{\beta}) = \int_{\boldsymbol{t}} p(\boldsymbol{t}|\alpha_0) \prod_n \sum_{z_n} p(w_n|\boldsymbol{\beta}_{z_n}) p(z_n|\boldsymbol{t}) d\boldsymbol{t} \,.$$

3 Topic Compositional Neural Language Model

We describe the proposed TCNLM, as illustrated in Figure 1. Our model consists of two key components:

(i) a neural topic model (NTM), and (ii) a neural language model (NLM). The NTM aims to capture the long-range semantic meanings across the document, while the NLM is designed to learn the local semantic and syntactic relationships between words.

3.1 Neural Topic Model

Let $\boldsymbol{d} \in \mathbb{Z}_+^D$ denote the bag-of-words representation of a document, with $\mathbb{Z}_+$ denoting nonnegative integers. D is the vocabulary size, and each element of $\boldsymbol{d}$ reflects a count of the number of times the corresponding word occurs in the document. Distinct from LDA (Blei et al., 2003), we pass a *Gaussian* random vector through a softmax function to parameterize the multinomial document topic distributions (Miao et al., 2017). Specifically, the generative process of the NTM is

$$\begin{aligned} \boldsymbol{\theta} &\sim \mathcal{N}(\mu_0, \sigma_0^2) & \boldsymbol{t} &= g(\boldsymbol{\theta}) \\ z_n &\sim \text{Discrete}(\boldsymbol{t}) & w_n &\sim \text{Discrete}(\boldsymbol{\beta}_{z_n})\,, \end{aligned} \tag{4}$$

where $\mathcal{N}(\mu_0, \sigma_0^2)$ is an isotropic Gaussian distribution, with mean μ_0 and variance σ_0^2 in each dimension; $g(\cdot)$ is a transformation function that maps sample $\boldsymbol{\theta}$ to the topic embedding $\boldsymbol{t}$, defined here as $g(\boldsymbol{\theta}) = \text{softmax}(\hat{\mathbf{W}}\boldsymbol{\theta} + \hat{\boldsymbol{b}})$, where $\hat{\mathbf{W}}$ and $\hat{\boldsymbol{b}}$ are trainable parameters.

The marginal likelihood for document $\boldsymbol{d}$ is:

$$\begin{aligned} p(\boldsymbol{d}|\mu_0, \sigma_0, \boldsymbol{\beta}) &= \int_{\boldsymbol{t}} p(\boldsymbol{t}|\mu_0, \sigma_0^2) \prod_n \sum_{z_n} p(w_n|\boldsymbol{\beta}_{z_n}) p(z_n|\boldsymbol{t}) d\boldsymbol{t} \\ &= \int_{\boldsymbol{t}} p(\boldsymbol{t}|\mu_0, \sigma_0^2) \prod_n p(w_n|\boldsymbol{\beta}, \boldsymbol{t}) d\boldsymbol{t} \\ &= \int_{\boldsymbol{t}} p(\boldsymbol{t}|\mu_0, \sigma_0^2) p(\boldsymbol{d}|\boldsymbol{\beta}, \boldsymbol{t}) d\boldsymbol{t}\,. \end{aligned} \tag{5}$$

The second equation in (5) holds because we can readily marginalized out the sampled topic words z_n by

$$p(w_n|\boldsymbol{\beta}, \boldsymbol{t}) = \sum_{z_n} p(w_n|\boldsymbol{\beta}_{z_n}) p(z_n|\boldsymbol{t}) = \boldsymbol{\beta}\boldsymbol{t}\,. \tag{6}$$

$\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_T\}$ is the transition matrix from the topic distribution to the word distribution, which are trainable parameters of the decoder; T is the number of topics and $\boldsymbol{\beta}_i \in \mathbb{R}^D$ is the topic distribution over words (all elements of $\boldsymbol{\beta}_i$ are nonnegative, and they sum to one).

The re-parameterization trick (Kingma and Welling, 2013) can be applied to build an unbiased and low-variance gradient estimator for the variational distribution. The parameter updates can still be derived directly from the variational lower bound, as discussed in Section 3.3.

Diversity Regularizer Redundance in inferred topics is a common issue exisiting in general topic models. In order to address this issue, it is straightforward to regularize the row-wise distance between each paired topics to diversify the topics. Following Xie et al. (2015); Miao et al. (2017), we apply a topic diversity regularization while carrying out the inference.

Specifically, the distance between a pair of topics are measured by their cosine distance $a(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) = \arccos\left(\frac{|\boldsymbol{\beta}_i \cdot \boldsymbol{\beta}_j|}{||\boldsymbol{\beta}_i||_2 ||\boldsymbol{\beta}_j||_2}\right)$. The mean angle of all pairs of T topics is $\phi = \frac{1}{T^2}\sum_i \sum_j a(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j)$, and the variance is $\nu = \frac{1}{T^2}\sum_i \sum_j (a(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) - \phi)^2$. Finally, the topic diversity regularization is defined as $R = \phi - \nu$.

3.2 Neural Language Model

We propose a Mixture-of-Experts (MoE) language model, which consists a set of "expert networks", *i.e.*, $E_1, E_2, ..., E_T$. Each expert is itself an RNN with its own parameters corresponding to a latent topic.

Without loss of generality, we begin by discussing an RNN with a simple transition function, which is then generalized to the LSTM. Specifically, we define two weight tensors $\mathcal{W} \in \mathbb{R}^{n_h \times n_x \times T}$ and $\mathcal{U} \in \mathbb{R}^{n_h \times n_h \times T}$, where n_h is the number of hidden units and n_x is the dimension of word embedding. Each expert E_k corresponds to a set of parameters $\mathcal{W}[k]$ and $\mathcal{U}[k]$, which denotes the k-th 2D "slice" of $\mathcal{W}$ and $\mathcal{U}$, respectively. All T experts work cooperatively to generate an output y_m. Sepcifically,

$$p(y_m) = \sum_{k=1}^{T} \boldsymbol{t}_k \cdot \text{softmax}(\mathbf{V}\boldsymbol{h}_m^{(k)}) \tag{7}$$

$$\boldsymbol{h}_m^{(k)} = \sigma(\mathcal{W}[k]\boldsymbol{x}_m + \mathcal{U}[k]\boldsymbol{h}_{m-1})\,, \tag{8}$$

where $\boldsymbol{t}_k$ is the usage of topic k (component k of $\boldsymbol{t}$), and $\sigma(\cdot)$ is a sigmoid function; $\mathbf{V}$ is the weight matrix connecting the RNN's hidden state, used for computing a distribution over words. Bias terms are omitted for simplicity.

However, such an MoE module is computationally prohibitive and storage excessive. The training process is inefficient and even infeasible in practice. To remedy this, instead of ensembling the output of the T experts as in (7), we extend the weight matrix of the RNN to be an ensemble of topic-dependent weight matrices. Specifically, the T experts work together as follows:

$$p(y_m) = \text{softmax}(\mathbf{V}\boldsymbol{h}_m) \tag{9}$$

$$\boldsymbol{h}_m = \sigma(\mathbf{W}(\boldsymbol{t})\boldsymbol{x}_m + \mathbf{U}(\boldsymbol{t})\boldsymbol{h}_{m-1})\,, \tag{10}$$

and

$$\mathbf{W}(\boldsymbol{t}) = \sum_{k=1}^{T} \boldsymbol{t}_k \cdot \mathcal{W}[k],\ \mathbf{U}(\boldsymbol{t}) = \sum_{k=1}^{T} \boldsymbol{t}_k \cdot \mathcal{U}[k]\,. \quad (11)$$

In order to reduce the number of model parameters, motivated by Gan et al. (2016); Song et al. (2016), instead of implementing a tensor as in (11), we decompose $\mathbf{W}(\boldsymbol{t})$ into a multiplication of three terms $\mathbf{W}_a \in R^{n_h \times n_f}$, $\mathbf{W}_b \in R^{n_f \times T}$ and $\mathbf{W}_c \in R^{n_f \times n_x}$, where n_f is the number of factors. Specifically,

$$\begin{aligned}\mathbf{W}(\boldsymbol{t}) &= \mathbf{W}_a \cdot \mathrm{diag}(\mathbf{W}_b \boldsymbol{t}) \cdot \mathbf{W}_c \\ &= \mathbf{W}_a \cdot (\mathbf{W}_b \boldsymbol{t} \odot \mathbf{W}_c)\,, \end{aligned} \quad (12)$$

where $\odot$ represents the Hadamard operator. $\mathbf{W}_a$ and $\mathbf{W}_c$ are shared parameters across all topics, to capture the common linguistic patterns. $\mathbf{W}_b$ are the factors which are weighted by the learned topic embedding $\boldsymbol{t}$. The same factorization is also applied for $\mathbf{U}(\boldsymbol{t})$. The topic distribution $\boldsymbol{t}$ affects RNN parameters associated with the document when predicting the succeeding words, which implicitly defines an ensemble of T language models. In this factorized model, the RNN weight matrices that correspond to each topic share "structure".

Now we generalize the above analysis by using LSTM units. Specifically, we summarize the new topic compositional LSTM cell as:

$$\begin{aligned}\boldsymbol{i}_m &= \sigma(\mathbf{W}_{ia}\tilde{\boldsymbol{x}}_{i,m-1} + \mathbf{U}_{ia}\tilde{\boldsymbol{h}}_{i,m-1}) \\ \boldsymbol{f}_m &= \sigma(\mathbf{W}_{fa}\tilde{\boldsymbol{x}}_{f,m-1} + \mathbf{U}_{fa}\tilde{\boldsymbol{h}}_{f,m-1}) \\ \boldsymbol{o}_m &= \sigma(\mathbf{W}_{oa}\tilde{\boldsymbol{x}}_{o,m-1} + \mathbf{U}_{oa}\tilde{\boldsymbol{h}}_{o,m-1}) \\ \tilde{\boldsymbol{c}}_m &= \sigma(\mathbf{W}_{ca}\tilde{\boldsymbol{x}}_{c,m-1} + \mathbf{U}_{ca}\tilde{\boldsymbol{h}}_{c,m-1}) \\ \boldsymbol{c}_m &= \boldsymbol{i}_m \odot \tilde{\boldsymbol{c}}_m + \boldsymbol{f}_m \cdot \boldsymbol{c}_{m-1} \\ \boldsymbol{h}_m &= \boldsymbol{o}_m \odot \tanh(\boldsymbol{c}_m)\,. \end{aligned} \quad (13)$$

For $* = i, f, o, c$, we define

$$\tilde{\boldsymbol{x}}_{*,m-1} = \mathbf{W}_{*b}\boldsymbol{t} \odot \mathbf{W}_{*c}\boldsymbol{x}_{m-1} \quad (14)$$

$$\tilde{\boldsymbol{h}}_{*,m-1} = \mathbf{U}_{*b}\boldsymbol{t} \odot \mathbf{U}_{*c}\boldsymbol{h}_{m-1}\,. \quad (15)$$

Compared with a standard LSTM cell, our LSTM unit has a total number of parameters in size of $4n_f \cdot (n_x + 2T + 3n_h)$ and the additional computational cost comes from (14) and (15). Further, empirical comparison has been conducted in Section 5.6 to verify that our proposed model is superior than using the naive MoE implementation as in (7).

3.3 Model Inference

The proposed model (see Figure 1) follows the variational autoencoder (Kingma and Welling, 2013) framework, which takes the bag-of-words as input and embeds a document into the topic vector. This vector is then used to reconstruct the bag-of-words input, and also to learn an ensemble of RNNs for predicting a sequence of words in the document.

The joint marginal likelihood can be written as:

$$p(y_{1:M}, \boldsymbol{d}|\mu_0, \sigma_0^2, \boldsymbol{\beta}) = \int_{\boldsymbol{t}} p(\boldsymbol{t}|\mu_0, \sigma_0^2) p(\boldsymbol{d}|\boldsymbol{\beta}, \boldsymbol{t}) \prod_{m=1}^{M} p(y_m|y_{1:m-1}, \boldsymbol{t}) d\boldsymbol{t}\,. \quad (16)$$

Since the direct optimization of (16) is intractable, we employ variational inference (Jordan et al., 1999). We denote $q(\boldsymbol{t}|\boldsymbol{d})$ to be the variational distribution for $\boldsymbol{t}$. Hence, we construct the variational objective function, also called the evidence lower bound (ELBO), as

$$\begin{aligned}\mathcal{L} &= \underbrace{\mathbb{E}_{q(\boldsymbol{t}|\boldsymbol{d})}\left(\log p(\boldsymbol{d}|\boldsymbol{t})\right) - \mathrm{KL}\left(q(\boldsymbol{t}|\boldsymbol{d})||p(\boldsymbol{t}|\mu_0, \sigma_0^2)\right)}_{\text{neural topic model}} \\ &+ \underbrace{\mathbb{E}_{q(\boldsymbol{t}|\boldsymbol{d})}\left(\sum_{m=1}^{M} \log p(y_m|y_{1:m-1}, \boldsymbol{t})\right)}_{\text{neural language model}} \\ &\leq \log p(y_{1:M}, \boldsymbol{d}|\mu_0, \sigma_0^2, \boldsymbol{\beta})\,. \end{aligned} \quad (17)$$

More details can be found in the Supplementary Material. In experiments, we optimize the ELBO together with the diversity regularisation:

$$\mathcal{J} = \mathcal{L} + \lambda \cdot R\,. \quad (18)$$

4 Related Work

Topic Model Topic models have been studied for a variety of applications in document modeling. Beyond LDA (Blei et al., 2003), significant extensions have been proposed, including capturing topic correlations (Blei and Lafferty, 2007), modeling temporal dependencies (Blei and Lafferty, 2006), discovering an unbounded number of topics (Teh et al., 2005), learning deep architectures (Henao et al., 2015; Zhou et al., 2015), among many others. Recently, neural topic models have attracted much attention, building upon the successful usage of restricted Boltzmann machines (Hinton and Salakhutdinov, 2009), autoregressive models (Larochelle and Lauly, 2012), sigmoid belief networks (Gan et al., 2015), and variational autoencoders (Miao et al., 2016).

Variational inference has been successfully applied in a variety of applications (Pu et al., 2016; Wang et al., 2017; Chen et al., 2017). The recent work of Miao et al. (2017) employs variational inference to train topic models, and is closely related to our work. Their model follows the original LDA formulation and extends it by parameterizing the multinomial distribution with neural networks. In contrast, our model

Dataset	Vocabulary		Training			Development			Testing		
	LM	TM	# Docs	# Sents	# Tokens	# Docs	# Sents	# Tokens	# Docs	# Sents	# Tokens
APNEWS	32,400	7,790	$50K$	$0.7M$	$15M$	$2K$	$27.4K$	$0.6M$	$2K$	$26.3K$	$0.6M$
IMDB	34,256	8,713	$75K$	$0.9M$	$20M$	$12.5K$	$0.2M$	$0.3M$	$12.5K$	$0.2M$	$0.3M$
BNC	41,370	9,741	$15K$	$0.8M$	$18M$	$1K$	$44K$	$1M$	$1K$	$52K$	$1M$

Table 1: Summary statistics for the datasets used in the experiments.

enforces the neural network not only modeling documents as bag-of-words, but also transfering the inferred topic knowledge to a language model for word-sequence generation.

Language Model Neural language models have recently achieved remarkable advances (Mikolov et al., 2010). The RNN-based language model (RNNLM) is superior for its ability to model longer-term temporal dependencies without imposing a strong conditional independence assumption; it has recently been shown to outperform carefully-tuned traditional n-gram-based language models (Jozefowicz et al., 2016).

An RNNLM can be further improved by utilizing the broad document context (Mikolov and Zweig, 2012). Such models typically extract latent topics via a topic model, and then send the topic vector to a language model for sentence generation. Important work in this direction include Mikolov and Zweig (2012); Dieng et al. (2016); Lau et al. (2017); Ahn et al. (2016). The key differences of these methods is in either the topic model itself or the method of integrating the topic vector into the language model. In terms of the topic model, Mikolov and Zweig (2012) uses a pre-trained LDA model; Dieng et al. (2016) uses a variational autoencoder; Lau et al. (2017) introduces an attention-based convolutional neural network to extract semantic topics; and Ahn et al. (2016) utilizes the topic associated to the fact pairs derived from a knowledge graph (Vinyals and Le, 2015).

Concerning the method of incorporating the topic vector into the language model, Mikolov and Zweig (2012) and Lau et al. (2017) extend the RNN cell with additional topic features. Dieng et al. (2016) and Ahn et al. (2016) use a hybrid model combining the predicted word distribution given by both a topic model and a standard RNNLM. Distinct from these approaches, our model learns the topic model and the language model jointly under the VAE framework, allowing an efficient end-to-end training process. Further, the topic information is used as guidance for a Mixture-of-Experts (MoE) model design. Under our factorization method, the model can yield boosted performance efficiently (as corroborated in the experiments).

Recently, Shazeer et al. (2017) proposes a MoE model for large-scale language modeling. Different from ours, they introduce a MoE layer, in which each expert stands for a small feed-forward neural network on the previous output of the LSTM layer. Therefore, it yields a significant quantity of additional parameters and computational cost, which is infeasible to train on a single GPU machine. Moreover, they provide no semantic meanings for each expert, and all experts are treated equally; the proposed model can generate meaningful sentences conditioned on given topics.

Our TCNLM is similar to Gan et al. (2016). However, Gan et al. (2016) uses a two-step pipline, first learning a multi-label classifier on a group of pre-defined image tags, and then generating image captions conditioned on them. In comparison, our model jointly learns a topic model and a language model, and focuses on the language modeling task.

5 Experiments

Datasets We present experimental results on three publicly available corpora: APNEWS, IMDB and BNC. APNEWS[1] is a collection of Associated Press news articles from 2009 to 2016. IMDB is a set of movie reviews collected by Maas et al. (2011), and BNC (BNC Consortium, 2007) is the written portion of the British National Corpus, which contains excerpts from journals, books, letters, essays, memoranda, news and other types of text. These three datasets can be downloaded from GitHub[2].

We follow the preprocessing steps in Lau et al. (2017). Specifically, words and sentences are tokenized using Stanford CoreNLP (Manning et al., 2014). We lowercase all word tokens, and filter out word tokens that occur less than 10 times. For topic modeling, we additionally remove stopwords[3] in the documents and exclude the top 0.1% most frequent words and also words that appear in less than 100 documents. All these datasets are divided into training, development and testing sets. A summary statistic of these datasets is provided in Table 1.

[1] https://www.ap.org/en-gb/

[2] https://github.com/jhlau/topically-driven-language-model

[3] We use the following stopwords list: https://github.com/mimno/Mallet/blob/master/stoplists/en.txt

Dataset	LSTM type	basic-LSTM	LDA+LSTM			LCLM	Topic-RNN			TCNLM		
			50	100	150		50	100	150	50	100	150
APNEWS	small	64.13	57.05	55.52	54.83	54.18	56.77	54.54	54.12	52.75	52.63	**52.59**
	large	58.89	52.72	50.75	50.17	50.63	53.19	50.24	50.01	48.07	47.81	**47.74**
IMDB	small	72.14	69.58	69.64	69.62	67.78	68.74	67.83	66.45	63.98	62.64	**62.59**
	large	66.47	63.48	63.04	62.78	67.86	63.02	61.59	60.14	57.06	56.38	**56.12**
BNC	small	102.89	96.42	96.50	96.38	87.47	94.66	93.57	93.55	87.98	86.44	**86.21**
	large	94.23	88.42	87.77	87.28	80.68	85.90	84.62	84.12	80.29	80.14	**80.12**

Table 2: Test perplexities of different language models on APNEWS, IMDB and BNC.

Setup For the NTM part, we consider a 2-layer feed-forward neural network to model $q(\boldsymbol{t}|\boldsymbol{d})$, with 256 hidden units in each layer; ReLU (Nair and Hinton, 2010) is used as the activation function. The hyper-parameter λ for the diversity regularizer is fixed to 0.1 across all the experiments. All the sentences in a paragraph, excluding the one being predicted, are used to obtain the bag-of-words document representation $\boldsymbol{d}$. The maximum number of words in a paragraph is set to 300.

In terms of the NLM part, we consider 2 settings: (i) a small 1-layer LSTM model with 600 hidden units, and (ii) a large 2-layer LSTM model with 900 hidden units in each layer. The sequence length is fixed to 30. In order to alleviate overfitting, dropout with a rate of 0.4 is used in each LSTM layer. In addition, adaptive softmax (Grave et al., 2016) is used to speed up the training process.

During training, the NTM and NLM parameters are jointly learned using Adam (Kingma and Ba, 2014). All the hyper-parameters are tuned based on the performance on the development set. We empirically find that the optimal settings are fairly robust across the 3 datasets. All the experiments were conducted using Tensorflow and trained on NVIDIA GTX TITAN X with 3072 cores and 12GB global memory.

5.1 Language Model Evaluation

Perplexity is used as the metric to evaluate the performance of the language model. In order to demonstrate the advantage of the proposed model, we compare TCNLM with the following baselines:

- **basic-LSTM**: A baseline LSTM-based language model, using the same architecture and hyper-parameters as TCNLM wherever applicable.
- **LDA+LSTM**: A topic-enrolled LSTM-based language model. We first pretrain an LDA model (Blei et al., 2003) to learn 50/100/150 topics for APNEWS, IMDB and BNC. Given a document, the LDA topic distribution is incorporated by concatenating with the output of the hidden states to predict the next word.
- **LCLM** (Wang and Cho, 2016): A context-based language model, which incorporates context information from preceding sentences. The preceding sentences are treated as bag-of-words, and an attention mechanism is used when predicting the next word. All hyper-parameters are set to be the same as in our TCNLM. The number of preceding sentences is tuned on the development set (4 in general).
- **Topic-RNN** (Dieng et al., 2016): A joint learning framework that learns a topic model and a language model simutaneously. The topic information is incorporated through a linear transformation to rescore the prediction of the next word.

Results are presented in Table 2. We highlight some observations. (i) All the topic-enrolled methods outperform the basic-LSTM model, indicating the effectiveness of incorporating global semantic topic information. (ii) Our TCNLM performs the best across all datasets, and the trend keeps improving with the increase of topic numbers. (iii) The improved performance of TCNLM over LCLM implies that encoding the document context into meaningful topics provides a better way to improve the language model compared with using the extra context words directly. (iv) The margin between LDA+LSTM/Topic-RNN and our TCNLM indicates that our model supplies a more efficient way to utilize the topic information through the joint variational learning framework to implicitly train an ensemble model.

5.2 Topic Model Evaluation

We evaluate the topic model by inspecting the coherence of inferred topics (Chang et al., 2009; Newman et al., 2010; Mimno et al., 2011). Following Lau et al. (2014), we compute topic coherence using normalized PMI (NPMI). Given the top n words of a topic, the coherence is calculated based on the sum of pairwise NPMI scores between topic words, where the word probabilities used in the NPMI calculation are based on co-occurrence statistics mined from English Wikipedia with a sliding window. In practice, we average topic coherence over the top 5/10/15/20

Dataset										
	army	**animal**	**medical**	**market**	**lottory**	**terrorism**	**law**	**art**	**transportation**	**education**
APNEWS	afghanistan	animals	patients	zacks	casino	syria	lawsuit	album	airlines	students
	veterans	dogs	drug	cents	mega	iran	damages	music	fraud	math
	soldiers	zoo	fda	earnings	lottery	militants	plaintiffs	film	scheme	schools
	brigade	bear	disease	keywords	gambling	al-qaida	filed	songs	conspiracy	education
	infantry	wildlife	virus	share	jackpot	korea	suit	comedy	flights	teachers
	horror	**action**	**family**	**children**	**war**	**detective**	**sci-fi**	**negative**	**ethic**	**epsiode**
IMDB	zombie	martial	rampling	kids	war	eyre	alien	awful	gay	season
	slasher	kung	relationship	snoopy	che	rochester	godzilla	unfunny	school	episodes
	massacre	li	binoche	santa	documentary	book	tarzan	sex	girls	series
	chainsaw	chan	marie	cartoon	muslims	austen	planet	poor	women	columbo
	gore	fu	mother	parents	jews	holmes	aliens	worst	sex	batman
	environment	**education**	**politics**	**business**	**facilities**	**sports**	**art**	**award**	**expression**	**crime**
BNC	pollution	courses	elections	corp	bedrooms	goal	album	john	eye	police
	emissions	training	economic	turnover	hotel	score	band	award	looked	murder
	nuclear	students	minister	unix	garden	cup	guitar	research	hair	killed
	waste	medau	political	net	situated	ball	music	darlington	lips	jury
	environmental	education	democratic	profits	rooms	season	film	speaker	stared	trail

Table 3: 10 topics learned from our TCNLM on APNEWS, IMDB and BNC.

# Topic	Model	Coherence		
		APNEWS	IMDB	BNC
50	LDA	0.125	0.084	0.106
	NTM	0.075	0.064	0.081
	Topic-RNN(s)	0.134	0.103	0.102
	Topic-RNN(l)	0.127	0.096	0.100
	TCNLM(s)	**0.159**	**0.106**	**0.114**
	TCNLM(l)	0.152	0.100	0.101
100	LDA	0.136	0.092	**0.119**
	NTM	0.085	0.071	0.070
	Topic-RNN(s)	0.158	0.096	0.108
	Topic-RNN(l)	0.143	0.093	0.105
	TCNLM(s)	**0.160**	**0.101**	0.111
	TCNLM(l)	0.152	0.098	0.104
150	LDA	0.134	0.094	**0.119**
	NTM	0.078	0.075	0.072
	Topic-RNN(s)	0.146	0.089	0.102
	Topic-RNN(l)	0.137	0.092	0.097
	TCNLM(s)	0.153	**0.096**	0.107
	TCNLM(l)	**0.155**	0.093	0.102

Table 4: Topic coherence scores of different models on APNEWS, IMDB and BNC. (s) and (l) indicate small and large model, respectively.

topic words. To aggregate topic coherence score for a trained model, we then further average the coherence scores over topics. For comparison, we use the following baseline topic models:

- **LDA**: LDA (Blei et al., 2003) is used as a baseline topic model. We use LDA to learn the topic distributions for LDA+LSTM.
- **NTM**: We evaluate the neural topic model proposed in Cao et al. (2015). The document-topic and topic-words multinomials are expressed using neural networks. N-grams embeddings are incorporated as inputs of the model.
- **Topic-RNN** (Dieng et al., 2016): The same model as used in the language model evaluation.

Results are summarized in Table 4. Our TCNLM achieves promising results. Specifically, (*i*) we achieve the best coherence performance over APNEWS and IMDB, and are relatively competitive with LDA on BNC. (*ii*) We also observe that a larger model may result in a slightly worse coherence performance. One possible explanation is that a larger language model may have more impact on the topic model, and the inherited stronger sequential information may be harmful to the coherence measurement. (*iii*) Additionally, the advantage of our TCNLM over Topic-RNN indicates that our TCNLM supplies a more powerful topic guidance.

In order to better understand the topic model, we provide the top 5 words for 10 randomly chosen topics on each dataset (the boldface word is the topic name summarized by us), as shown in Table 3. These results correspond to the small network with 100 neurons. We also present some inferred topic distributions for several documents from our TCNLM in Figure 2. The topic usage for a specific document is sparse, demonstrating the effectiveness of our NTM. More inferred topic distribution examples are provided in the Supplementary Material.

5.3 Sentence Generation

Another advantage of our TCNLM is its capacity to generate meaningful sentences conditioned on given topics. Given topic i, we construct an LSTM generator by using only the i-th factor of $\mathbf{W}_b$ and $\mathbf{U}_b$. Then we start from a zero hidden state, and greedily sample words until an end token occurs. Table 5 shows the generated sentences from our TCNLM learned with 50 topics using the small network. Most of the sentences are strongly correlated with the given topics. More interestingly, we can also generate reasonable sentences conditioned on a mixed combination of topics, even if the topic pairs are divergent, *e.g.*, "an-

Data	Topic	Generated Sentences
APNEWS	army	• a female sergeant, serving in the fort worth, has served as she served in the military in iraq .
	animal	• most of the bear will have stumbled to the lake .
	medical	• physicians seeking help in utah and the nih has had any solutions to using the policy and uses offline to be fitted with a testing or body .
	market	• the company said it expects revenue of $ <unk> million to $ <unk> million in the third quarter .
	lottory	• where the winning numbers drawn up for a mega ball was sold .
	army+terrorism	• the taliban 's presence has earned a degree from the 1950-53 korean war in pakistan 's historic life since 1964 , with two example of <unk> soldiers from wounded iraqi army shootings and bahrain in the eastern army .
	animal+lottory	• she told the newspaper that she was concerned that the buyer was in a neighborhood last year and had a gray wolf .
IMDB	horror	• the killer is a guy who is n't even a zombie .
	action	• the action is a bit too much , but the action is n't very good .
	family	• the film is also the story of a young woman whose <unk> and <unk> and very yet ultimately sympathetic , <unk> relationship , <unk> , and palestine being equal , and the old man , a <unk> .
	children	• i consider this movie to be a children 's film for kids .
	war	• the documentary is a documentary about the war and the <unk> of the war .
	horror+negative	• if this movie was indeed a horrible movie i think i will be better off the film .
	sci-fi+children	• paul thinks him has to make up when the <unk> eugene discovers defeat in order to take too much time without resorting to mortal bugs , and then finds his wife and boys .
BNC	environment	• environmentalists immediate base calls to defend the world .
	education	• the school has recently been founded by a <unk> of the next generation for two years .
	politics	• a new economy in which privatization was announced on july 4 .
	business	• net earnings per share rose <unk> % to $ <unk> in the quarter , and $ <unk> m , on turnover that rose <unk> % to $ <unk> m.
	facilities	• all rooms have excellent amenities .
	environment+politics	• the commission 's report on oct. 2 , 1990 , on jan. 7 denied the government 's grant to " the national level of water " .
	art+crime	• as well as 36, he is returning freelance into the red army of drama where he has finally been struck for their premiere .

Table 5: Generated sentences from given topics. More examples are provided in the Supplementary Material.

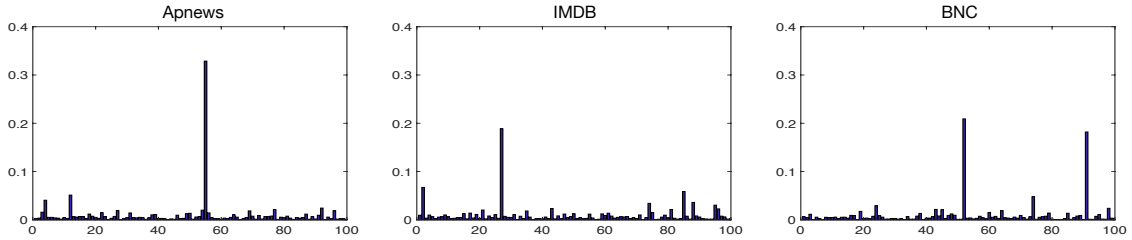

Figure 2: Inferred topic distributions on one sample document in each dataset. Content of the three documents is provided in the Supplementary Mateiral.

imal" and "lottory" for APNEWS. More examples are provided in the Supplementary Material. It shows that our TCNLM is able to generate topic-related sentences, providing an interpretable way to understand the topic model and the language model simulaneously. These qualitative analysis further demonstrate that our model effectively assembles the meaning of topics to generate sentences.

5.4 Empirical Comparison with Naive MoE

We explore the usage of a naive MoE language model as in (7). In order to fit the model on a single GPU machine, we train a NTM with 30 topics and each NLM of the MoE is a 1-layer LSTM with 100 hidden units. Results are summarized in Table 6. Both the naive MoE and our TCNLM provide better performance than the basic LSTM. Interestingly, though requiring less computational cost and storage usage, our TCNLM outperforms the naive MoE by a non-trivial margin. We attribute this boosted performance to the "structure" design of our matrix factorization method. The inherent topic-guided factor control significantly prevents overfitting, and yields efficient training, demonstrating the advantage of our model for transferring semantic knowledge learned from the topic model to the language model.

Dataset	basic-LSTM	naive MoE	TCNLM
APNEWS	101.62	85.87	**82.67**
IMDB	105.29	96.16	**94.64**
BNC	146.50	130.01	**125.09**

Table 6: Test perplexity comparison between the naive MoE implementation and our TCNLM on APNEWS, IMDB and BNC.

6 Conclusion

We have presented Topic Compositional Neural Language Model (TCNLM), a new method to learn a topic model and a language model simultaneously. The topic model part captures the global semantic meaning in a document, while the language model part learns the local semantic and syntactic relationships between words. The inferred topic information is incorporated into the language model through a Mixture-of-Experts model design. Experiments conducted on three corpora validate the superiority of the proposed approach. Further, our model infers sensible topics, and has the capacity to generate meaningful sentences conditioned on given topics. One possible future direction is to extend the TCNLM to a conditional model and apply it for the machine translation task.

References

S. Ahn, H. Choi, T. Pärnamaa, and Y. Bengio. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*, 2016.

E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran. Deep neural network language models. In *NAACL-HLT Workshop*, 2012.

D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, 2006.

D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 2007.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

B. BNC Consortium. The british national corpus, version 3 (bnc xml edition). *Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium. URL:http://www.natcorp.ox.ac.uk/*, 2007.

Z. Cao, S. Li, Y. Liu, W. Li, and H. Ji. A novel neural topic model and its supervised extension. In *AAAI*, 2015.

J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

C. Chen, C. Li, L. Chen, W. Wang, Y. Pu, and L. Carin. Continuous-time flows for deep generative models. *arXiv preprint arXiv:1709.01179*, 2017.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell. Language models for image captioning: The quirks and what works. *arXiv preprint arXiv:1505.01809*, 2015.

A. B. Dieng, C. Wang, J. Gao, and J. Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*, 2016.

Z. Gan, C. Chen, R. Henao, D. Carlson, and L. Carin. Scalable deep poisson factor analysis for topic modeling. In *ICML*, 2015.

Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng. Semantic compositional networks for visual captioning. *arXiv preprint arXiv:1611.08002*, 2016.

É. Grave, A. Joulin, M. Cissé, D. Grangier, and H. Jégou. Efficient softmax approximation for gpus. *arXiv preprint arXiv:1609.04309*, 2016.

R. Henao, Z. Gan, J. Lu, and L. Carin. Deep poisson factor modeling. In *NIPS*, 2015.

G. E. Hinton and R. R. Salakhutdinov. Replicated softmax: an undirected topic model. In *NIPS*, 2009.

S. Hochreiter and J. Schmidhuber. Long short-term memory. In *Neural computation*, 1997.

Y. H. Hu, S. Palreddy, and W. J. Tompkins. A patient-adaptable ecg beat classifier using a mixture of experts approach. *IEEE transactions on biomedical engineering*, 1997.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 1999.

R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

H. Larochelle and S. Lauly. A neural autoregressive topic model. In *NIPS*, 2012.

J. H. Lau, D. Newman, and T. Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL*, 2014.

J. H. Lau, T. Baldwin, and T. Cohn. Topically driven neural language model. *arXiv preprint arXiv:1704.08012*, 2017.

A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *ACL*, 2011.

C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL*, 2014.

J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

Y. Miao, L. Yu, and P. Blunsom. Neural variational inference for text processing. In *ICML*, 2016.

Y. Miao, E. Grefenstette, and P. Blunsom. Discovering discrete latent topics with neural variational inference. *arXiv preprint arXiv:1706.00359*, 2017.

T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. *SLT*, 2012.

T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, 2010.

D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, 2011.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *NAACL*, 2010.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

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 *NIPS*, 2016.

H. Schwenk, A. Rousseau, and M. Attik. Large, pruned or continuous space language models on a gpu for statistical machine translation. In *NAACL-HLT Workshop*, 2012.

N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

J. Song, Z. Gan, and L. Carin. Factored temporal sigmoid belief networks for sequence learning. In *ICML*, 2016.

A. Sriram, H. Jun, S. Satheesh, and A. Coates. Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426*, 2017.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *NIPS*, 2005.

A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang. Decoding with large-scale neural language models improves translation. In *EMNLP*, 2013.

O. Vinyals and Q. Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

L. Wan, L. Zhu, and R. Fergus. A hybrid neural network-latent topic model. In *AISTAT*, 2012.

T. Wang and H. Cho. Larger-context language modelling with recurrent neural network. *ACL*, 2016.

W. Wang, Y. Pu, V. K. Verma, K. Fan, Y. Zhang, C. Chen, P. Rai, and L. Carin. Zero-shot learning via class-conditioned deep generative models. *arXiv preprint arXiv:1711.05820*, 2017.

P. Xie, Y. Deng, and E. Xing. Diversifying restricted boltzmann machine for document modeling. In *KDD*, 2015.

M. Zhou, Y. Cong, and B. Chen. The poisson gamma belief network. In *NIPS*, 2015.