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 pretrained 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
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 quantifies the probability of topic usage within a document. Latent code is further used in a Mixture-of-Experts model (Hutb, 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 the 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 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 probability of each word \( y_m \) given all the previous words \( y_{1:m-1} \) through the hidden state \( h_m \):

\[
p(y_m | y_{1:m-1}) = p(y_m | h_m) \tag{2}
\]

\[
h_m = f(h_{m-1}, x_m). \tag{3}
\]

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), provides a robust and scalable approach for document modeling, by introducing latent variables for each token, indicating its topic assignment. Specifically, let \( t \) 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 \( z_n \):

\[
t \sim \text{Dir}(\alpha_0), z_n \sim \text{Discrete}(t) , w_n \sim \text{Discrete}(\beta_{z_n}),
\]

where \( \beta_{z_n} \) represents the distribution over words for topic \( z_n \); \( \alpha_0 \) 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

\[
p(d | \alpha_0, \beta) = \int_p(t | \alpha_0) \prod_n z_n p(w_n | \beta_{z_n}) p(z_n | t) dt.
\]

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 \( d \in \mathbb{Z}_+^D \) denote the bag-of-words representation of a document, with \( \mathbb{Z}_+ \) denoting nonnegative integers. \( D \) is the vocabulary size, and each element of \( d \) 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

\[
\theta \sim \mathcal{N}(\mu_0, \sigma_0^2) \quad t = g(\theta) \\
z_n \sim \text{Discrete}(t) \quad w_n \sim \text{Discrete}(\beta_{z_n}),
\]

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

The marginal likelihood for document \( d \) is:

\[
p(d|\mu_0, \sigma_0, \beta) = \int_{t} p(t|\mu_0, \sigma_0^2) \prod_{n} \sum_{z_n} p(w_n|\beta_{z_n})p(z_n|t)dt \\
= \int_{t} p(t|\mu_0, \sigma_0^2) \prod_{n} p(w_n|\beta_t)dt \\
= \int_{t} p(t|\mu_0, \sigma_0^2)p(d|\beta_t)dt.
\]

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

\[
p(w_n|\beta, t) = \sum_{z_n} p(w_n|\beta_{z_n})p(z_n|t) = \beta t.
\]

\( \beta = \{\beta_1, \beta_2, ..., \beta_T\} \) 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 \( \beta_i \in \mathbb{R}^D \) is the topic distribution over words (all elements of \( \beta_i \) 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 existing 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 regularizer while carrying out the inference.

Specifically, the distance between a pair of topics are measured by their cosine distance

\[
a(\beta_i, \beta_j) = \arccos \left( \frac{\beta_i \cdot \beta_j}{\|\beta_i\|_2 \|\beta_j\|_2} \right)
\]

The mean angle of all pairs of \( T \) topics is \( \phi = \frac{1}{T^2} \sum_i \sum_j a(\beta_i, \beta_j) \), and the variance is \( \nu = \frac{1}{T^2} \sum_i \sum_j (a(\beta_i, \beta_j) - \phi)^2 \). 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 \( W \in \mathbb{R}^{n_h \times n_h \times T} \) and \( U \in \mathbb{R}^{n_h \times n_h \times T} \), 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 \( W \) and \( U \), respectively. All \( T \) experts work cooperatively to generate an output \( y_m \). Specifically,

\[
p(y_m) = \sum_{k=1}^{T} t_k \cdot \text{softmax}(\mathbf{V} h_m^{(k)})
\]

\[
h_m^{(k)} = \sigma(\mathcal{W}[k] x_m + \mathcal{U}[k] h_{m-1}),
\]

where \( t_k \) is the usage of topic \( k \) (component \( k \) of \( t \)), 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:

\[
p(y_m) = \text{softmax}(\mathbf{V} \mathbf{h}_m)
\]

\[
\mathbf{h}_m = \sigma(\mathbf{W}(t) x_m + \mathbf{U}(t) \mathbf{h}_{m-1}),
\]
and
\[ W(t) = \sum_{k=1}^{T} t_k \cdot W[k], \quad U(t) = \sum_{k=1}^{T} t_k \cdot U[k]. \] (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 \( W(t) \) into a multiplication of three terms \( W_a \in \mathbb{R}^{n_h \times n_f} \), \( W_b \in \mathbb{R}^{n_f \times T} \) and \( W_c \in \mathbb{R}^{n_f \times n_v} \), where \( n_f \) is the number of factors. Specifically,
\[
W(t) = W_a \cdot \text{diag}(W_b t) \cdot W_c
= W_a \cdot (W_b t \odot W_c),
\] (12)

where \( \odot \) represents the Hadamard operator. \( W_a \) and \( W_c \) are shared parameters across all topics, to capture the common linguistic patterns. \( W_b \) are the factors which are weighted by the learned topic embedding \( t \). The same factorization is also applied for \( U(t) \).

The topic distribution \( t \) 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:
\[
i_m = \sigma(W_{ia} \hat{x}_{i,m-1} + U_{ia} \hat{h}_{i,m-1})
\]
\[
f_m = \sigma(W_{fa} \hat{x}_{f,m-1} + U_{fa} \hat{h}_{f,m-1})
\]
\[
o_m = \sigma(W_{oa} \hat{x}_{o,m-1} + U_{oa} \hat{h}_{o,m-1})
\]
\[
\tilde{c}_m = \sigma(W_{ca} \hat{x}_{c,m-1} + U_{ca} \hat{h}_{c,m-1})
\]
\[
c_m = i_m \odot \tilde{c}_m + f_m \cdot c_{m-1}
\]
\[
h_m = o_m \odot \text{tanh}(c_m).
\] (13)

For \( * = i, f, o, c \), we define
\[
\hat{x}_{*,m-1} = W_{s*} t \odot W_{s*} x_{m-1}
\] (14)
\[
\hat{h}_{*,m-1} = U_{s*} t \odot U_{s*} h_{m-1}.
\] (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:
\[
p(y_{1:M}, d | \mu_0, \sigma_0^2, \beta) = \int p(t | \mu_0, \sigma_0^2) p(d | \beta, t) \prod_{m=1}^{M} p(y_m | y_{1:m-1}, t) dt.
\] (16)

Since the direct optimization of (16) is intractable, we employ variational inference (Jordan et al., 1999). We denote \( q(t | d) \) to be the variational distribution for \( t \). Hence, we construct the variational objective function, also called the evidence lower bound (ELBO), as
\[
\mathcal{L} = \mathbb{E}_{q(t|d)} \left( \log p(d | t) - \text{KL} (q(t|d) || p(t | \mu_0, \sigma_0^2)) \right) + \mathbb{E}_{q(t|d)} \left( \sum_{m=1}^{M} \log p(y_m | y_{1:m-1}, t) \right)
\]
\[
\leq \log p(y_{1:M}, d | \mu_0, \sigma_0^2, \beta).
\] (17)

More details can be found in the Supplementary Material. In experiments, we optimize the ELBO together with the diversity regularisation:
\[
\mathcal{J} = \mathcal{L} + \lambda \cdot R.
\] (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...
enforces the neural network not only modeling documents as bag-of-words, but also transferring 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 (Jozeowicz 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 TCNLML is similar to Gan et al. (2016). However, Gan et al. (2016) uses a two-step pipeline, 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 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, mem- oranda, news and other types of text. These three datasets can be downloaded from GitHub.

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 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.

![Table 1: Summary statistics for the datasets used in the experiments.](https://github.com/jhlau/topically-driven-language-model)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Vocabulary</th>
<th>LM</th>
<th>Tex</th>
<th># Docs</th>
<th># Sents</th>
<th># Tokens</th>
<th>Development</th>
<th># Docs</th>
<th># Sents</th>
<th># Tokens</th>
<th>Testing</th>
<th># Docs</th>
<th># Sents</th>
<th># Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>APNEWS</td>
<td>32,400</td>
<td>7,790</td>
<td>50K</td>
<td>0.7M</td>
<td>15M</td>
<td>2K</td>
<td>27.4K</td>
<td>0.6M</td>
<td>2K</td>
<td>26.3K</td>
<td>0.6M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMDB</td>
<td>34,256</td>
<td>8,713</td>
<td>75K</td>
<td>0.9M</td>
<td>20M</td>
<td>12.5K</td>
<td>0.2M</td>
<td>0.3M</td>
<td>12.5K</td>
<td>0.2M</td>
<td>0.3M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNC</td>
<td>41,370</td>
<td>9,741</td>
<td>15K</td>
<td>0.8M</td>
<td>18M</td>
<td>1K</td>
<td>44K</td>
<td>1M</td>
<td>1K</td>
<td>52K</td>
<td>1M</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. We use the following stopwords list: https://github.com/mimno/Mallet/blob/master/stoplists/en.txt
Table 2: Test perplexities of different models on APNEWS, IMDB, and BNC. (* ) taken from Lau et al. (2017).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>LSTM type</th>
<th>basic-LSTM*</th>
<th>LDA+LSTM*</th>
<th>LCLM*</th>
<th>Topic-RNN</th>
<th>TDL^*</th>
<th>TCNLM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>100</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>APNEWS</td>
<td>small</td>
<td>64.13</td>
<td>57.05</td>
<td>55.52</td>
<td>54.83</td>
<td>54.18</td>
<td>56.74</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>58.89</td>
<td>52.72</td>
<td>50.75</td>
<td>50.17</td>
<td>50.63</td>
<td>53.19</td>
</tr>
<tr>
<td>IMDB</td>
<td>small</td>
<td>72.14</td>
<td>69.58</td>
<td>69.64</td>
<td>69.62</td>
<td>61.78</td>
<td>68.74</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>66.47</td>
<td>63.48</td>
<td>63.04</td>
<td>62.78</td>
<td>67.80</td>
<td>63.02</td>
</tr>
<tr>
<td>BNC</td>
<td>small</td>
<td>102.59</td>
<td>90.42</td>
<td>90.59</td>
<td>86.35</td>
<td>87.47</td>
<td>94.66</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>94.23</td>
<td>88.42</td>
<td>87.77</td>
<td>87.28</td>
<td>80.68</td>
<td>85.90</td>
</tr>
</tbody>
</table>

Setup: For the NTM part, we consider a 2-layer feed-forward neural network to model $q(t|d)$, with 256 hidden units in each layer; ReLU (Nair and Hinton, 2010) is used as the activation function. The hyperparameter $\lambda$ 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 $d$. 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 hyperparameters 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.
- **TDL^* (Lau et al., 2017)**: A convolutional topic enrolled language model. Its topic knowledge is utilized by concatenating to a dense layer of a recurrent language model.
- **Topic-RNN** (Dieng et al., 2016): A joint learning framework that learns a topic model and a language model simultaneously. The topic information is incorporated through a linear transformation to rescore the prediction of the next word.

Topic-RNN (Dieng et al., 2016) is implemented by ourselves and other comparisons are copied from (Lau et al., 2017). 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,
Table 4: Topic coherence scores of different models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>army</th>
<th>animal</th>
<th>medical</th>
<th>market</th>
<th>lottery</th>
<th>terrorism</th>
<th>law</th>
<th>art</th>
<th>transportation</th>
<th>education</th>
</tr>
</thead>
<tbody>
<tr>
<td>APNEWS</td>
<td>Afghan</td>
<td>animal</td>
<td>market</td>
<td>lottery</td>
<td>terrorism</td>
<td>law</td>
<td>art</td>
<td>transportation</td>
<td>education</td>
<td></td>
</tr>
<tr>
<td></td>
<td>veterans</td>
<td>dogs</td>
<td>drug</td>
<td>cents</td>
<td>mega</td>
<td>iran</td>
<td>damages</td>
<td>music</td>
<td>fraud</td>
<td>math</td>
</tr>
<tr>
<td>IMDB</td>
<td>zombie</td>
<td>martial</td>
<td>ramping</td>
<td>kite</td>
<td>war</td>
<td>eye</td>
<td>alen</td>
<td>awful</td>
<td>gay</td>
<td>season</td>
</tr>
<tr>
<td></td>
<td>slasher</td>
<td>kung</td>
<td>relationship</td>
<td>snoopy</td>
<td>che</td>
<td>rochester</td>
<td>godzilla</td>
<td>unfunny</td>
<td>school</td>
<td>episodes</td>
</tr>
<tr>
<td>BNC</td>
<td>pollution</td>
<td>courses</td>
<td>elections</td>
<td>corp</td>
<td>bedrooms</td>
<td>goal</td>
<td>album</td>
<td>john</td>
<td>eye</td>
<td>police</td>
</tr>
<tr>
<td></td>
<td>emissions</td>
<td>training</td>
<td>economic</td>
<td>turnover</td>
<td>hotel</td>
<td>score</td>
<td>band</td>
<td>award</td>
<td>looked</td>
<td>murder</td>
</tr>
<tr>
<td></td>
<td>nuclear</td>
<td>students</td>
<td>minister</td>
<td>unix</td>
<td>garden</td>
<td>cup</td>
<td>guitar</td>
<td>research</td>
<td>hair</td>
<td>killed</td>
</tr>
<tr>
<td></td>
<td>waste</td>
<td>medias</td>
<td>political</td>
<td>net</td>
<td>situated</td>
<td>ball</td>
<td>music</td>
<td>darlington</td>
<td>lips</td>
<td>jury</td>
</tr>
</tbody>
</table>

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

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

Table 4: Topic coherence scores of different models on APNEWS, IMDB and BNC. (s) and (l) indicate small and large model, respectively. (*) taken from Lau et al. (2017).

we average topic coherence over the top 5/10/15/20

Sentence Generation

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 \(W_b \) and \(U_b\). 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. Additionally, the advantage of our TCNLM over Topic-RNN indicates that our TCNLM supplies a more powerful topic guidance.
Topic Compositional Neural Language Model

<table>
<thead>
<tr>
<th>Data</th>
<th>Topic</th>
<th>Generated Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>APNEWS</td>
<td>army</td>
<td>• A female sergeant, serving in the Iraq war, has served as a police officer in Iraq.</td>
</tr>
<tr>
<td></td>
<td>animal</td>
<td>• Most of the bears have been trained to the lake.</td>
</tr>
<tr>
<td></td>
<td>medical</td>
<td>• Physicians seeking help in Utah and the NH has had any solutions to turn the policy and use offline to be fitted with a testing or body.</td>
</tr>
<tr>
<td></td>
<td>market</td>
<td>• The company said it expects revenue of $&lt;unk&gt; million to $&lt;unk&gt; million in the third quarter.</td>
</tr>
<tr>
<td></td>
<td>lottery</td>
<td>• The winning numbers drawn for the Mega Ball were sold.</td>
</tr>
<tr>
<td></td>
<td>army+terror</td>
<td>• The Taliban's offensive in Pakistan has been labeled as the most since 1973, with two examples of $&lt;unk&gt;$.</td>
</tr>
<tr>
<td></td>
<td>animal-lottery</td>
<td>• She told the newspaper that she was concerned that the buyer was in neighborhood last year and had a gray wolf.</td>
</tr>
<tr>
<td>IMDB</td>
<td>horror</td>
<td>• The killer is a guy who is n't even a zombie.</td>
</tr>
<tr>
<td></td>
<td>action</td>
<td>• The action is a hit too much, but the action is n't very good.</td>
</tr>
<tr>
<td></td>
<td>family</td>
<td>• The film is also the story of a young woman whose &lt;unk&gt; and &lt;unk&gt; and very yet ultimately sympathetic, &lt;unk&gt; relationship, &lt;unk&gt;, and palestine being equal, and the old man, a &lt;unk&gt;.</td>
</tr>
<tr>
<td></td>
<td>children</td>
<td>• I consider this movie to be a children's film for kids.</td>
</tr>
<tr>
<td></td>
<td>war</td>
<td>• The documentary is a documentary about the war and the &lt;unk&gt; of the war.</td>
</tr>
<tr>
<td></td>
<td>horror+negative</td>
<td>• If the movie was indeed a horrible movie I think I will be better off the film.</td>
</tr>
<tr>
<td></td>
<td>sea-f+children</td>
<td>• Paul thinks him has to make up when the &lt;unk&gt; has to make up what the &lt;unk&gt; has to make up when the &lt;unk&gt; has to make up what the &lt;unk&gt; has to make up.</td>
</tr>
<tr>
<td>BNC</td>
<td>environment</td>
<td>• Environmentalists immediate base calls to defend the world.</td>
</tr>
<tr>
<td></td>
<td>politics</td>
<td>• The school has recently been founded by a &lt;unk&gt; of the next generation for two years.</td>
</tr>
<tr>
<td></td>
<td>business</td>
<td>• Net earnings per share rose &lt;unk&gt; % to $&lt;unk&gt; in the quarter, and $&lt;unk&gt; m, on growth that rose &lt;unk&gt; % to $&lt;unk&gt; m.</td>
</tr>
<tr>
<td></td>
<td>art+crime</td>
<td>• As well as 36, he is returning freelance into the red army of drama where he has finally been struck for their premiere.</td>
</tr>
</tbody>
</table>

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 Material.

ics, even if the topic pairs are divergent, e.g., “animal” and “lottery” 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 simultaneously. 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>basic-LSTM</th>
<th>naive MoE</th>
<th>TCNLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>APNEWS</td>
<td>101.62</td>
<td>85.87</td>
<td>82.67</td>
</tr>
<tr>
<td>IMDB</td>
<td>105.29</td>
<td>94.64</td>
<td>94.64</td>
</tr>
<tr>
<td>BNC</td>
<td>146.50</td>
<td>130.01</td>
<td>125.09</td>
</tr>
</tbody>
</table>

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**


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.


