Revisiting Topic-Guided Language Models

Anonymous authors
Paper under double-blind review

Abstract

A recent line of work in natural language processing has aimed to combine language models and topic models. These *topic-guided language models* augment neural language models with topic models, unsupervised learning methods that can discover document-level patterns of word use. This paper compares the effectiveness of these methods in a standardized setting. We study four topic-guided language models and two baselines, evaluating the held-out predictive performance of each model on three corpora. Surprisingly, we find that *none of these methods outperform a standard LSTM language model baseline*, and most fail to learn good topics. Further, we train a probe of the neural language model that shows that the baseline's hidden states already encode topic information. We make public all code used for this study.

1 Introduction

Recurrent neural networks (RNNs) and LSTMs have been an important class of models in the development of methods for many tasks in natural language processing, including machine translation, summarization, and speech recognition. One of the most successful applications of these models is in language modeling, where they are effective at modeling small text corpora. Even with the advent of transformer-based language models, RNNs and LSTMs can outperform non-pretrained transformers on various small datasets (Melis et al., 2020).

While powerful, RNN- and LSTM-based models struggle to capture long-range dependencies in their context history (Bai et al., 2018; Sankar et al., 2019). Additionally, they are not designed to learn interpretable structure in a corpus of documents. To this end, multiple researchers have proposed adapting these models by incorporating topic models (Dieng et al., 2016; Lau et al., 2017; Rezaee & Ferraro, 2020; Guo et al., 2020). The motivation for combining language models and topic models is to decouple local syntactic structure, which can be modeled by a language model, from document-level semantic concepts, which can be captured by a topic model (Khandelwal et al., 2018; O'Connor & Andreas, 2021). The topic model component is also designed to uncover latent structure in documents.

We refer to these models as *topic-guided language models*. Broadly, this body of research has reported good results: topic-guided language models improve next-word predictive performance and learn interpretable topics.

In this work, we re-investigate this class of models by evaluating four representative topic-guided language model (TGLM) papers in a unified setting. We train the models from Dieng et al. (2016); Lau et al. (2017); Rezaee & Ferraro (2020); Guo et al. (2020) on three document-level corpora and evaluate their held-out perplexity. Unlike some prior work, during next-word prediction, we take care to condition the topic model component on only previous words, rather than the entire document. Moreover, we use a baseline language model that is conditioned on *all* previously seen document words, rather than being restricted to the current sentence (Lau et al., 2017; Rezaee & Ferraro, 2020; Guo et al., 2020). Additionally, we choose baseline language models with comparable model sizes to ensure valid comparisons. Our finding: no predictive improvement of TGLMs over a standard LSTM-LM baseline (Zaremba et al., 2014).

In order to understand why topic-guided language models offer no predictive improvement, we probe the LSTM-LM's hidden representations. A probe is a trained predictor used to measure the extent to which fitted "black-box" models, such as neural models, have learned specific linguistic features of the input (Hewitt

& Liang, 2019). The probe reveals that the LSTM-LM already encodes topic information, rendering a formal topic model component redundant.

Additionally, topic-guided language models were developed to provide insight into text corpora by uncovering latent topics. This method of exploratory text analysis is commonly used in the social sciences and digital humanities (Griffiths & Steyvers, 2004; Blei & Lafferty, 2007; Grimmer & Stewart, 2013; Mohr & Bogdanov, 2013). Here, we show that the topics learned by topic-guided language models are not better than a standard topic model and, for some of the models, qualitatively poor.

This paper contributes to a line of reproducibility studies in machine learning that aim to evaluate competing methods in a consistent and equitable manner. These studies have uncovered instances where results are not directly comparable, as reported numbers are borrowed from prior works that used different experimental settings (Marie et al., 2021; Hoyle et al., 2021). Furthermore, they identify cases where baselines are either too weak or improperly tuned (Dacrema et al., 2019). We observe analogous issues within the topic-guided language modeling literature. To support transparency and reproducibility, we will make public all code used in this study.

Finally, we consider how these insights apply to other models. While prior work has incorporated topic model components into RNNs and LSTMs, the topic-guided language model framework is agnostic to the class of neural language model used. We conclude by discussing how the results in this paper are relevant to researchers considering incorporating topic models into more powerful neural language models, such as transformers.

2 Study Design

Let $x_{1:T} = \{x_1, \ldots, x_T\}$ be a sequence of tokens collectively known as a document, where each x_t indexes one of V words in a vocabulary (words outside the vocabulary are mapped to a special out-of-vocabulary token). Given a corpus of documents, the goal of language modeling is to learn a model $p(x_{1:T})$ that approximates the probability of observing a document.

A document can be modeled autoregressively using the chain rule of probability,

$$p(x_{1:T}) = \textstyle\prod_{t=1}^{T} p(x_t \mid x_{<t}), \tag{1}$$

where $x_{<t}$ denotes all the words in a document before t. A language model parameterizes the predictive distribution of the next word, $p_\mu(x_t \mid x_{<t})$, with a set of parameters μ. Given a set of documents indexed by D_{train}, we compute a parameter estimate $\hat{\mu}$ by maximizing the log likelihood objective,

$$\textstyle\sum_{d=1}^{D_{\text{train}}} \sum_{t=1}^{T_d} \log p_\mu(x_{d,t} \mid x_{d,<t}),$$

with respect to μ. Language models are evaluated using perplexity on a held-out set of documents. With D_{test} as the index set of the test documents, perplexity is defined as

$$\exp\left\{-\frac{1}{\sum_d T_d} \textstyle\sum_{d=1}^{D_{\text{test}}} \sum_{t=1}^{T_d} \log p_{\hat{\mu}}(x_{d,t} \mid x_{d,<t})\right\}.$$

Perplexity is the inverse geometric average of the likelihood of observing each word in the set of test documents under the fitted model; a lower perplexity indicates a better model fit.

3 Language Models and Topic Models

Here, we provide an overview of the two components of topic-guided language models: neural language models and topic models. The topic-guided language model literature has focused on models based on RNNs and LSTMs, which are the neural language models we focus on here.

RNN language model. A recurrent neural network (RNN) language model (Mikolov et al., 2010) defines each conditional probability in Equation (1) as

$$h_{t-1} = f(x_{t-1}, h_{t-2}) \tag{2}$$

$$p_{\text{RNN}}(x_t \mid x_{<t}) = \text{softmax}(\mathbf{W}^\intercal h_{t-1}), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{D \times V}$ and $h_t \in \mathbb{R}^D$. The hidden state h_{t-1} summarizes the information in the preceding sequence, while the function f combines h_{t-1} with the word at time t to produce a new hidden state, h_t. The function f is parameterized by a recurrent neural network (RNN).

The parameter $\mathbf{W}$ and the RNN model parameters are trained by maximizing the log likelihood of training documents using backpropagation through time (BPTT) (Williams & Peng, 1990). (In practice, the backpropagation of gradients is truncated after a specified sequence length.) The model directly computes the predictive distribution of the next word, $p(x_t \mid x_{<t})$.

The baselines use the widely used RNN architecture, the LSTM (Hochreiter & Schmidhuber, 1997), as the language model, which we call LSTM-LM (Zaremba et al., 2014). The LSTM architecture is described in Appendix A.

To make full use of the document context, it is natural to condition on all previous words of the document when computing $p(x_t \mid x_{<t})$. Even when the full document does not fit into memory, this can be done at no extra computational cost by memoizing the previous word's hidden state (h_{t-2} in Equation (2)) (Melis et al., 2017). This is our main baseline.

One can also define $x_{<t}$ to be only the previous words in the current sentence. In this scenario, the model will not condition on all prior words in the document. This is the LSTM-LM baseline used in many TGLM papers (Lau et al., 2017; Guo et al., 2020; Rezaee & Ferraro, 2020). We call this model the sentence-level LSTM-LM.

Topic model. Another way to model a document is with a bag-of-words model that represents documents as word counts. One such model is a probabilistic topic model, which assumes the observed words are conditionally independent given a latent variable θ. In a topic model, the probability of a document is

$$p(x_{1:T}) = \int \prod_{i=1}^{T} p(x_i \mid \theta) p(\theta) d\theta, \tag{4}$$

where $p(\theta)$ is a prior distribution on θ and $p(x \mid \theta)$ is the likelihood of word x conditional on θ.

A widely used probabilistic topic model is Latent Dirichlet Allocation (LDA) (Blei et al., 2003). LDA posits that a corpus of text is comprised of K latent topics. Each document d contains a distribution over topics, θ_d, and each topic k is associated with a distribution over words, β_k. These two terms combine to form the distribution of each word in a document.

The generative model for LDA is:

1. Draw K topics: $\beta_1, \ldots, \beta_K \sim \text{Dirichlet}_V(\gamma)$.
2. For each document:
   (a) Draw topic proportions,
   $\theta \sim \text{Dirichlet}_K(\alpha)$.
   (b) For each word $x_1, \ldots, x_T$:
      i. Draw topic indicator,
      $z_{x_t} \sim \text{Categorical}(\theta)$.
      ii. Draw word, $x_t \sim \text{Categorical}(\beta_{z_{x_t}})$.

Since each word is drawn conditionally independent of the preceding words in the document, LDA is not able to capture word order or syntax. However, it can capture document-level patterns since the topic for each word is drawn from a document-specific distribution. Practitioners typically rely on approximate posterior inference to estimate the LDA posterior. The most common methods are Gibbs sampling (Griffiths & Steyvers, 2004) or variational inference (Blei et al., 2003).

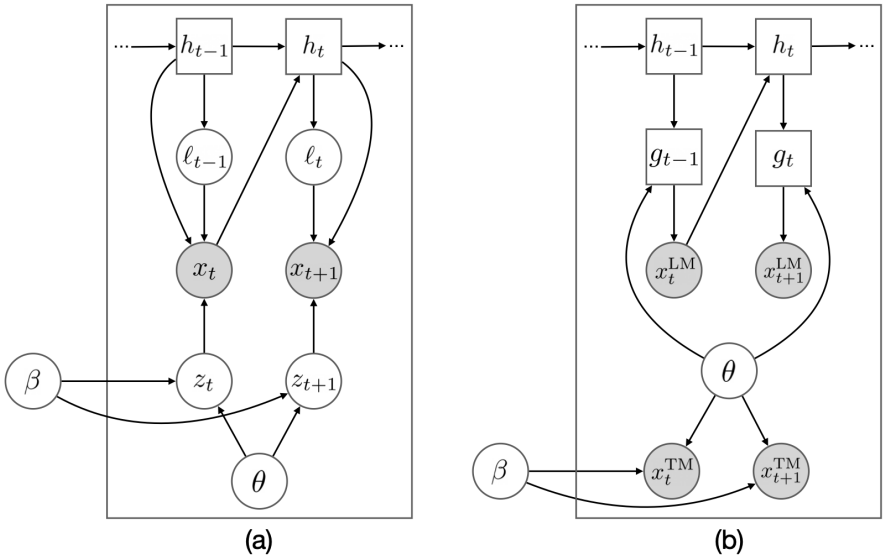

Figure 1: Graphical model representations of the two frameworks of topic-guided language models. (a) is the topic-biased language model. (b) is the joint topic and language model. Circles denote random variables, while squares denote deterministic variables. Shading indicates that the variable is observed.

After approximating the posterior distribution over topics from the training documents, the next-word posterior predictive distribution is

$$p_{\text{LDA}}(x_t \mid x_{<t}) = \int p(x_t \mid \theta) p(\theta \mid x_{<t}) d\theta. \tag{5}$$

Given words $x_{<t}$ from a document, one can use approximate posterior inference to estimate the topic proportions posterior, $p(\theta \mid x_{<t})$, and then draw Monte Carlo samples of θ to estimate the predictive distribution.

4 Topic-Guided Language Model

We now discuss topic-guided language models (TGLMs), which are a class of language models that combine topic models and neural language models. TGLMs were initially proposed to combine the fluency of neural language models with the document modeling capabilities of topic models. Dieng et al. (2016) and Lau et al. (2017), who propose two of the models that we study here, argue that long-range dependency in language is captured well by topic models. Subsequent TGLM papers build on Dieng et al. (2016) and Lau et al. (2017), but differ from these previous works in evaluation setting (Wang et al., 2018; Rezaee & Ferraro, 2020; Guo et al., 2020).

Topic-guided language models can be divided into two frameworks, differing in whether they model the document's bag-of-words counts in addition to the typical next-word prediction objective. In this section, we discuss the two frameworks: a topic-biased language model and a joint topic and language model. The graphical structure of these models are shown in Figure 1.

4.1 Topic-Biased Language Models

A topic-biased language model defines the next-word probability to be the sum of two terms: a linear transformation of the hidden state, as in an RNN, and the distribution of words according to a document's topics, as in a topic model.

Each document follows the data generating mechanism below:

1. Draw topic vector, $\theta \sim \text{Dirichlet}_K(\cdot)$.

2. For each word $x_1, \ldots, x_T$:
   (a) $h_t = \text{RNN}(x_t, h_{t-1})$.
   (b) Draw $\ell_t \sim \text{Bernoulli}(\sigma(u^\intercal h_t))$.
   (c) Draw $z_t \sim \text{Categorical}(\theta)$.
   (d) Draw $x_{t+1} \propto \exp(\mathbf{W}^\intercal h_t + (1 - \ell_t)\beta_{z_t})$.

Here, $\sigma(\cdot)$ denotes the sigmoid function. The model parameters are the parameters of the RNN, the weights $\mathbf{W} \in \mathbb{R}^{D \times V}$ and $u \in \mathbb{R}^D$, and the topics $\beta_1, \ldots, \beta_K \in \mathbb{R}^V$. The latent variable θ determines the document's topic proportions.

Of the two additive terms in a word's likelihood, the RNN term encourages fluency and syntax while the topic modeling term can be understood as a bias toward the document's global structure. Since topic models struggle with modeling very common words ("stop words") (Wallach et al., 2009), a word's likelihood only includes the topic modeling term if it is not predicted to be a stop word ($\ell_t = 0$). The realizations of the stop word indicators are observed during training ($\ell_t = 1$ if x_{t+1} belongs to a list of stop words, and 0 otherwise). During prediction, the stop word indicators are treated as latent variables and are marginalized out.

TopicRNN. TopicRNN (Dieng et al., 2016) approximates Step 2(d) by marginalizing z_t before normalizing:

$$\begin{aligned} p_{\text{TRNN}}(x_{t+1} \mid h_t, \theta) &\propto \exp(\mathbb{E}[\mathbf{W}^\intercal h_t + (1 - \ell_t)\beta_{z_t}]) \\ &= \exp(\mathbf{W}^\intercal h_t + (1 - \ell_t)\beta^\intercal \theta). \end{aligned}$$

The topic matrix $\beta \in \mathbb{R}^{K \times V}$ contains the topic vectors $\beta_1, \ldots, \beta_K$ as rows. Additionally, in Step 1, TopicRNN draws θ from a standard Gaussian, rather than a Dirichlet distribution.

VRTM. VRTM (Rezaee & Ferraro, 2020) (short for Variational Recurrent Topic Model) exactly computes Step 2(d) by marginalizing z_t after normalizing:

$$\begin{aligned} p_{\text{VRTM}}(x_{t+1} \mid h_t, \theta) &= \mathbb{E}[\text{softmax}(\mathbf{W}^\intercal h_t + (1 - \ell_t)\beta_{z_t})] \\ &= \textstyle\sum_{k=1}^{K} \theta_k * \text{softmax}(\mathbf{W}^\intercal h_t + (1 - \ell_t)\beta_k). \end{aligned}$$

This makes VRTM a mixture-of-RNNs (Yang et al., 2017), where the mixture proportions are determined by θ.

Inference. The model parameters for topic-biased language models are learned using variational inference (Wainwright et al., 2008; Blei et al., 2017). We provide a high-level overview of the method here.

The goal of variational inference is to approximate the posterior of the latent variable θ, $p(\theta \mid x_{1:T})$, with a learned distribution $q_\phi(\theta)$, called the variational distribution. To fit $q_\phi(\theta)$, variational inference minimizes the KL divergence between the two distributions. This is equivalent to maximizing a lower bound of the marginal log likelihood, or evidence lower bound (ELBO):

$$\begin{aligned} \log p(x_{1:T}) &= \log \textstyle\int p_\mu(x_{1:T} \mid \theta) p(\theta) d\theta \\ &\geq \mathbb{E}_{q_\phi(\theta)}[\log p_\mu(x_{1:T} \mid \theta)] - \text{KL}(q_\phi(\theta) \| p(\theta)). \end{aligned}$$

The ELBO contains two terms: a reconstruction loss, which is the expected log probability of the data under $q_\phi(\theta)$, and the KL divergence between the variational distribution and the prior on θ. By maximizing the ELBO, we can simultaneously learn the model parameters and the variational distribution parameters, represented by μ and ϕ respectively. In order to share the learned variational parameters, the variational distribution is defined to be a function of the data, i.e., we learn $q_\phi(\theta \mid x_{1:T})$, where q_ϕ is parameterized by a neural network.

In practice, the ELBO is maximized with respect to parameters μ and ϕ using backpropagation. The expectation is estimated using samples from $q_\phi(\theta \mid x_{1:T})$ and the KL can often be computed analytically (e.g., when both distributions are Gaussians) (Kingma & Welling, 2013).

Prediction. For both models, the next-word predictive distribution is

$$p(x_t \mid x_{<t}) = \mathbb{E}_{p(\theta \mid x_{<t})}[p(x_t \mid x_{<t}, \theta)]. \tag{6}$$

Using a learned variational distribution in place of the exact posterior, we approximate the expectation using its mean, i.e., let $\hat{\theta} = \mathbb{E}[q_\phi(\theta \mid x_{<t})]$. Then $p(x_t \mid x_{<t}) \approx p(x_t \mid x_{<t}, \hat{\theta})$. For computation reasons, in practice, $\hat{\theta}$ is only updated in a sliding window (i.e., every N words).

4.2 Joint Topic and Language Model

A joint topic and language model learns the topic model and language model simultaneously, essentially fitting two views of the same data. The two models share the document-level latent variable θ.

Each document during training has a pair of representations, its bag-of-words $x_{1:T}^{\text{TM}}$ and its word sequence $x_{1:T}^{\text{LM}}$, generated by the topic model and the language model, respectively. For each document, a basic version of the data generating mechanism is:

1. Draw topic vector, $\theta \sim \text{Dirichlet}_K(\cdot)$.
2. Draw the bag-of-words $x_{1:T}^{\text{TM}}$ from a topic model (Section 3):
   (a) $x_{1:T}^{\text{TM}} \sim \text{TopicModel}(\theta)$.
3. For each word $x_1^{\text{LM}}, \ldots, x_T^{\text{LM}}$:
   (a) $h_t = \text{RNN}(x_t^{\text{LM}}, h_{t-1})$.
   (b) $g_t = a(h_t, \theta)$.
   (c) Draw $x_{t+1}^{\text{LM}} \propto \exp(\mathbf{W}^\intercal g_t)$.

Here, the latent variable θ determines the document's topic proportions in the topic model. In the language model, the hidden state h_t is combined with θ in a differentiable function a, usually the Gated Recurrent Unit (Cho et al., 2014). The GRU architecture is described in Appendix B.

The model parameters are the parameters of the topic model, the parameters of the RNN, the parameters of a, and the weights $\mathbf{W} \in \mathbb{R}^{D \times V}$.

TDLM. TDLM (Lau et al., 2017) (short for Topically Driven Language Model) is a variant of the model outlined above. There are two major differences. First, θ is not considered to be a latent variable. Instead, an encoder function maps a bag-of-words to θ. In the topic model, the bag-of-words used is from the entire document, i.e., $\theta^{\text{TM}} = \text{enc}(x_{1:T}^{\text{TM}})$.

Second, the language model component of the data generating process (Step 3) uses a different θ than the topic modeling component, which we call θ^{LM}. To prevent the model from memorizing the current sentence, θ^{LM} is computed from the document bag-of-words excluding the current sentence. In the language model, if j is the index set of the words in the current sentence, $\theta^{\text{LM}} = \text{enc}(x_{1:T \setminus j}^{\text{TM}})$.

Inference. TDLM is trained by maximizing the log likelihood of the topic model and the language model jointly. The objective, $\mathcal{L}_{\text{TDLM}}$, is:

$$\begin{aligned} \mathcal{L}_{\text{TM}} &= \log p(x_{1:T}^{\text{TM}} \mid \theta^{\text{TM}}) \\ \mathcal{L}_{\text{LM}} &= \textstyle\sum_t \log p(x_t^{\text{LM}} | x_{<t}^{\text{LM}}, \theta^{\text{LM}}) \\ \mathcal{L}_{\text{TDLM}} &= \mathcal{L}_{\text{TM}} + \mathcal{L}_{\text{LM}}. \end{aligned}$$

Although the original model only conditions on previous words in the current sentence when forming $\mathcal{L}_{\text{LM}}$, we condition on all prior words in the document because it improves performance.

Prediction. Let $\hat{\theta} = \text{enc}(x_{<t}^{\text{TM}})$. The next-word predictive distribution is

$$p(x_t^{\text{LM}} \mid x_{<t}^{\text{LM}}) = p(x_t^{\text{LM}} \mid x_{<t}^{\text{LM}}, \hat{\theta}), \tag{7}$$

which is defined in Step 3 of the data generating process. In practice, like prediction for the topic-biased LMs, we recompute $\hat{\theta}$ in a sliding window.

rGBN-RNN. rGBN-RNN (Guo et al., 2020) is an extension of the model outlined in this section. In rGBN-RNN's topic model, each sentence j has a unique bag-of-words: it is the document's bag-of-words with the sentence excluded, denoted $x^{\text{TM}}_{1:T\setminus j}$. In Step 1 of the data generating mechanism, a different topic vector is drawn sequentially for each sentence. For sentences $j = 1, \ldots, J$, where J is the total number of sentences,

$$\theta_j \sim \text{Gamma}(\Pi\theta_{j-1}, \tau_0),$$

where Π and τ_0 are model parameters. In Step 2, for each sentence j, its bag-of-words is drawn:

$$x^{\text{TM}}_{1:T\setminus j} \sim \text{Poisson}(\Phi\theta_j),$$

where Φ is a model parameter.

For the language modeling component (Step 3 of the data generating mechanism), rGBN-RNN generates individual sentences. In Step 3, each sentence j is conditionally independent of the other sentences, given its corresponding topic vector, θ_j. In other words,

$$p(x^{\text{LM}}_{j_t}|x^{\text{LM}}_{<j_t}, \theta_j) = p(x^{\text{LM}}_{j_t}|x^{\text{LM}}_{j,<t}, \theta_j), \tag{8}$$

where $x^{\text{TM}}_{j_t}$ is the t'th word of sentence j, $x^{\text{LM}}_{<j_t}$ denotes all the words in document before the t'th word of the j'th sentence, and $x^{\text{LM}}_{j,<t}$ denotes only the words in the j'th sentence before the t'th word.

rGBN-RNN also introduces multiple stochastic layers to both the topic model and language model, which is simplified to one layer in this exposition. In the experiments, we use the full original model.

Inference. rGBN-RNN is trained using a combination of variational inference and stochastic gradient MCMC (Guo et al., 2018). We refer the reader to Guo et al. (2020) for further mathematical details of the model and inference algorithm.

Prediction. For each sentence j, the next-word predictive distribution is

$$p(x^{\text{LM}}_{j,t} \mid x^{\text{LM}}_{<j_t}) = \mathbb{E}_{p(\theta_j \mid x^{\text{TM}}_{<j_1})}[p(x^{\text{LM}}_{j,t} \mid x^{\text{LM}}_{j,<t}, \theta_j)].$$

The expectation is approximated using a sample from the approximate posterior of θ_j computed during inference. The topic model parameters, Φ and Π, are similarly marginalized out via MCMC sampling (see Guo et al. (2020) for more details).

5 Experiments

In this section, we detail the reproducibility study and results. We also investigate the quality of learned topics and probe the LSTM-LM's hidden representations to demonstrate the amount of topic information being retained.

5.1 Reproducibility Study

We evaluate the held-out perplexity of two LSTM-LM baselines and four TGLMs on three document-level corpora.

Datasets. We use three publicly available natural language datasets: APNEWS,[1] IMDB (Maas et al., 2011), and BNC (Consortium, 2007). We follow the training, validation, and test splits from Lau et al. (2017). Details about the datasets and preprocessing steps are in Appendix C.

Models. The LSTM-LM is described in Section 2. The four topic-guided language models are TDLM (Lau et al., 2017), TopicRNN (Dieng et al., 2016), rGBN-RNN (Guo et al., 2020), and VRTM (Rezaee & Ferraro, 2020), and are described in Section 4.

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

Table 1: The LSTM-LM baseline matches or exceeds topic-guided language models in held-out perplexity. In parentheses are standard errors estimated by averaging three runs with different random seeds. The comparable baseline to each topic-guided language model is the LSTM-LM with the same number of parameters.

				Perplexity		
Model	LSTM Size	Topic Size	# Param.	APNEWS	IMDB	BNC
LSTM-LM (sentence-level)	600	–	2.2M	65.7 (0.1)	77.9 (0.1)	114.7 (0.2)
LSTM-LM (Zaremba et al., 2014)	600	–	2.2M	55.1 (0.1)	71.5 (0.1)	95.4 (0.4)
	600 (+GRU)	–	3.3M	52.5 (0.2)	66.9 (0.1)	91.8 (0.1)
	600x3	–	7.9M	50.5 (0.5)	64.6 (0.8)	87.3 (0.4)
TopicRNN (Dieng et al., 2016)	600	100	2.8M	55.8 (0.2)	72.1 (0.2)	96.3 (0.2)
VRTM (Rezaee & Ferraro, 2020)	600	50	3.2M	57.4 (0.6)	75.3 (0.3)	98.5 (0.6)
TDLM (Lau et al., 2017)	600	100	3.3M	53.8 (0.4)	67.5 (0.1)	90.9 (0.4)
rGBN-RNN (Guo et al., 2020)	600x3	100-80-50	11.6M	52.6 (0.3)	64.8 (0.2)	97.7 (1.1)

We implement TopicRNN (Dieng et al., 2016), TDLM (Lau et al., 2017), and VRTM (Rezaee & Ferraro, 2020) from scratch. For rGBN-RNN (Guo et al., 2020), we use the publicly available codebase and make minimal adjustments to the codebase to ensure that preprocessing and evaluation are consistent. Some other topic-guided language models do not have public code (Wang et al., 2018; Tang et al., 2019; Wang et al., 2019) and are not straightforward to implement. These models are not part of the study, but their architecture is similar to that of Lau et al. (2017), which we compare to.

For all LSTM-LM baselines, we use a hidden size of 600, word embeddings of size 300 initialized with Google News word2vec embeddings (Mikolov et al., 2013), and dropout of 0.4 between the LSTM input and output layers (and between the hidden layers for the 3-layer models). For the four TGLMs we study, we use the same settings as LSTM-LM for the LSTM components. For the additional TGLM-specific components, we use the architectures and settings from the original papers, except for small details reported in Appendix D to make certain settings consistent across models.[2]

To obtain comparable baselines to all TGLMs studied, we train three LSTM-LMs of varying sizes. The default baseline is a 1-layer LSTM-LM. To control for the additional GRU layer in the language model component of TDLM, we train a 1-layer LSTM-LM with a GRU layer between the LSTM output and the output embedding layer. To compare to rGBN-RNN, a hierarchical model, we train a 3-layer LSTM-LM. Finally, we also compare to a baseline considered in prior work, an LSTM-LM conditioned only on previous words in the same sentence. We call this model the sentence-level LSTM-LM.

Training Details. We train the RNN components using truncated backpropagation through time with a sequence length of 30. For sequences within the same document (or sentence for the sentence-level LSTM-LM), if h_i is the final hidden state computed by the RNN for the ith sequence, we initialize the initial hidden state for the $(i+1)^{\text{th}}$ sequence with stop_gradient(h_i). Although the TDLM and rGBN-RNN models assume conditional independence between sentences, we found that in practice, keeping hidden states between sentences improved their performance.

We use the Adam optimizer with a learning rate of 0.001. The models are trained until validation loss does not improve for 10 epochs and we use the checkpoint with the best validation perplexity. The models in our codebase train to convergence within three days on a single TITAN V GPU. rGBN-RNN, trained using its public codebase, trains to convergence within one week on the same GPU.

Results. Table 1 shows the results. After controlling for language model size, *the LSTM-LM baseline consistently matches or outperforms topic-guided language models with the same number of parameters.*

[2]The one additional change is that Guo et al. (2020) reports a 600-512-256 size model, but their public code only supports 3-layer models with same-size RNN layers, so we use 600-600-600.

Table 2: The probe experiment reveals that the hidden state of an LSTM-LM is predictive of TDLM's topic information on held-out data.

Metric	APNEWS	IMDB	BNC
Accuracy (Top-1)	.520	.456	.408
Accuracy (Top-5)	.890	.720	.887
R^2	.601	.427	.257

Although most topic-guided language models improve over a baseline considered in prior work — the sentence level LSTM-LM — they are matched by an LSTM which, like topic-guided language models, conditions on all prior words in a document. As discussed in Section 2, this is a standard practice that can be performed at no extra computational cost.

5.2 Probing the RNN hidden states

The motivation for topic-guided language models is to augment language models with document-level information captured by topic models (Dieng et al., 2016). To assess whether topic models are adding useful information, we perform a probe experiment. In NLP, probe tasks are designed to understand the extent to which linguistic structure is encoded in the representations produced by black-box models (Alain & Bengio, 2016; Conneau et al., 2018; Hewitt & Liang, 2019; Pimentel et al., 2020). In this case, we probe the baseline LSTM-LM's hidden representations to assess how much topic information it already captures.

Specifically, we evaluate whether an LSTM-LM's hidden representation of the document's first t words, h_t, is predictive of the topic vector estimated from the document's first t words, θ_t, of a topic-guided language model. We use TDLM as the topic-guided language model, since it is the best performing topic-guided language model and has the highest quality topics (see Section 5.3).

We use the fitted models from Section 5.1 to create training data for the probe experiment. The input is the LSTM-LM's initial hidden state h_t for each sequence in a document (in this experiment, we define a sequence to be a 30-word chunk). The output is TDLM's topic proportions at the sequence, transformed with inverse-softmax to ensure it is real-valued: $\tilde{\theta}_t = \log(\theta_t) - \sum_{j=1}^{K} \log(\theta_{t,j})$, where j indexes the dimension of θ.

In the experiment, a linear model is trained to predict $\tilde{\theta}_t$ from h_t. The loss function is mean squared error summed across the topic components. The held-out prediction quality of this model (or "probe") can be viewed as a proxy for mutual information between the LSTM-LM's hidden state and the topic proportion vector θ, which is not easily estimable.

Table 2 shows the results of the probe experiment. The linear model can reconstruct TDLM's topic proportion vector to some extent; 60% of the variance in a held-out set of APNEWS topic proportions can be explained by the LSTM's hidden state. Moreover, the hidden state can predict the top-5 accuracy of the largest topic more than 70% of the time across datasets. These high accuracies indicate that the LSTM-LM captures the most prevelant topic for the majority of sequences.

A caveat to analyzing these results is that a probe's reconstruction performance depends on the extent to which topics are helpful for language modeling. For example, the magnitudes of the R^2 values have the same ordering as TDLM's topic coherences (see Section 5.3), indicating that the predictiveness of the probe is correlated with TDLM's topic quality. Nevertheless, the probing task shows that a notable amount of broader topic information is captured by the baseline LSTM's hidden state.

5.3 Topic quality

Although the predictions from topic-guided language models are matched or exceeded by an LSTM-LM baseline, it is possible that the learned topics will still be useful to practitioners. We compare the topics

Table 3: Topic coherences across corpora and models. The topic-guided language models do not learn topics that are more coherent than LDA. Note we do not include VRTM in the table because the model did not learn distinct topics (see Table 5).

	Coherence		
Model	APNEWS	IMDB	BNC
LDA	.143	.092	.135
TopicRNN	.021	.024	.024
TDLM	.201	.098	.079
rGBN-RNN	.060	.019	.053

learned by topic-guided language models to those learned by a classical topic model, LDA (Blei et al., 2003). While LDA's next word predictions are worse than those of neural topic models (Dieng et al., 2020), its topics may be more interpretable. To assess the quality of learned topics, we compute an automated metric, topic coherence, that correlates with human judgements (Aletras & Stevenson, 2013; Lau et al., 2014).

Topic coherence is computed using normalized pointwise mutual information (NPMI) scores. The NPMI score of a topic from its N top words is defined as

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-\log p(w_i, w_j)}. \tag{9}$$

We use the corpus containing the training data to compute the word co-occurrence statistics, i.e., estimates of $p(w_i)$ and $p(w_i, w_j)$ in Equation (9). To obtain the model-level coherence, we average the coherences over the top 5/10/15/20 topic words for each topic, then average over all topics.[3]

The coherence for each model is in Table 3. The topic-biased language models (TopicRNN and VRTM) learn largely incoherent topics, while only TDLM (Lau et al., 2017) achieves comparable coherences to LDA across corpora. Since the quality of automated topic evaluation metrics like coherence has been disputed for neural topic models (Hoyle et al., 2021), Appendix E includes the top words from randomly sampled topics. The joint topic and language models (TDLM and rGBN-RNN) learn qualitatively acceptable topics, while VRTM fails to learn distinct topics entirely.

6 Discussion

This reproducibility study compares topic-guided language models to LSTM baselines. We find that the baselines outperform the topic-guided language models in predictive performance. This finding differs from the results reported in the topic-guided language modeling literature, which shows improvements over baselines. In general, these differences are due to weaker baselines in the literature and a form of evaluation that considers future words.

Baselines. The baseline compared to in most prior work (Lau et al., 2017; Rezaee & Ferraro, 2020; Guo et al., 2020) is the sentence-level LSTM-LM, which we report as the weakest model in Table 1; this baseline does not condition on all words in a document's history. Similarly, the baseline in Dieng et al. (2016) does not condition on the history beyond a fixed-context window. In contrast, the LSTM-LM baseline in this work conditions on all previous words in the document during training and evaluation. Our findings suggest that the the predictive advantage of topic-guided language models stems from conditioning on all prior words in a document via a representation of topic proportions.

Additionally, topic-guided language models typically augment their language model component with additional parameters. We only compare topic-guided language models to baselines with a similar number of

[3]We use the topic interpretability toolkit from Lau et al. (2014), `https://github.com/jhlau/topic_interpretability`.

language model parameters. TDLM (Lau et al., 2017) adds parameters to its language model via an additional GRU; however, TDLM was originally compared to a baseline without this module. We find that TDLM's predictive advantage fades when scaling the LSTM-LM baseline to match TDLM's language model size.

Evaluation. Different papers have evaluated the performance of topic-guided language models in different ways. In this paper, we standardize the evaluation of models and their baselines to make sure results are comparable.

As described in Section 4, topic-guided language models use a representation of the full document to estimate the topic proportions vector θ during training. During evaluation, θ must be estimated using only previous document words. Otherwise, a model would be looking ahead when making next-word predictions.[4]

Some methods in the TGLM literature have conditioned on future words in their evaluation, making them incomparable to language models that only condition on previous words. For example, TDLM (Lau et al., 2017) is proposed as a sentence-level model, and thus the paper reports results when conditioning on future words. Additionally, VRTM (Rezaee & Ferraro, 2020) and rGBN-RNN (Guo et al., 2020) are proposed as document-level models, but in the respective evaluation scripts of their public codebases, θ is estimated using future words.

Finally, conditioning during evaluation isn't the only difference among these models. Some prior papers do not use consistent language model vocabulary sizes, which makes reported numbers incomparable. For example, VRTM employs a smaller vocabulary size than the baselines and other models it compares to. These discrepancies in evaluation may account for differences in reported results.

7 Conclusion

We find that compared to a standard LSTM language model, topic-guided language models do not improve language modeling performance. The probe experiment shows that this is due to both the standard LSTM already possessing some level of topic understanding but also that capturing the exact topic vector is unnecessary for the task. In models without an explicit topic model, the quality of the learned topics is poor.

While this study shows that current topic-guided language models do not improve next-word predictive performance, it is possible that incorporating a topic model can provide greater control or diversity in language model generations. A promising future direction is adapting current topic-guided language models to these use cases.

The topic-guided language modeling literature has focused on LSTMs, but we note that this framework is agnostic to the class of neural language model used. This means the same framework can be used to incorporate topic models into more powerful neural language models, such as transformers (Vaswani et al., 2017). However, if incorporating topic models into transformers does not improve predictive performance or provide meaningful latent variables, it is not necessarily because of architectural differences between transformers and LSTMs. Rather, the probing results in this paper indicate that neural language models are sufficiently expressive such that they already retain topic information. Transformers, which are more expressive than LSTMs, are likely even more capable of capturing topic information without explicitly modeling topics. Future work can prioritize developing methods that allow expressive neural language models and interpretable, topic-based latent variables to be learned jointly.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[4] We also ran experiments that corrected this mismatch by estimating θ using only previous document words in both training and evaluation, but found that this did not help performance.

Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics*, pp. 13–22, 2013.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. doi: 10.48550/ARXIV.1803.01271. URL https://arxiv.org/abs/1803.01271.

David M. Blei and John D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, October 2014. doi: 10.3115/v1/W14-4012. URL https://aclanthology.org/W14-4012.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2126–2136, July 2018. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

BNC Consortium. The British National Corpus, version 3 (BNC XML Edition). Distributed by Oxford University Computing Services on behalf of the BNC Consortium., 2007. URL http://www.natcorp.ox.ac.uk/.

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 101–109, 2019. ISBN 9781450362436. doi: 10.1145/3298689.3347058. URL https://doi.org/10.1145/3298689.3347058.

Adji B. Dieng, Chong Wang, Jianfeng Gao, and John Paisley. TopicRNN: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*, 2016.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 2020. doi: 10.1162/tacl_a_00325. URL https://aclanthology.org/2020.tacl-1.29.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl_1):5228–5235, 2004. doi: 10.1073/pnas.0307752101. URL https://www.pnas.org/doi/abs/10.1073/pnas.0307752101.

Justin Grimmer and Brandon M Stewart. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3):267–297, 2013.

Dandan Guo, Bo Chen, Hao Zhang, and Mingyuan Zhou. Deep Poisson Gamma dynamical systems. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/file/4ffb0d2ba92f664c2281970110a2e071-Paper.pdf.

Dandan Guo, Bo Chen, Ruiying Lu, and Mingyuan Zhou. Recurrent hierarchical topic-guided RNN for language generation. In *International Conference on Machine Learning*, pp. 3810–3821, 2020.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. Is automated topic model evaluation broken? The incoherence of coherence. *Advances in Neural Information Processing Systems*, 34, 2021.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 423–430, 2003.

Jey H. Lau, David Newman, and Timothy Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *European Chapter of the Association for Computational Linguistics*, pp. 530–539, 2014.

Jey H. Lau, Timothy Baldwin, and Trevor Cohn. Topically driven neural language model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 355–365, 2017.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, June 2011. URL http://www.aclweb.org/anthology/P11-1015.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 7297–7306, August 2021. doi: 10.18653/v1/2021.acl-long.566. URL https://aclanthology.org/2021.acl-long.566.

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. *CoRR*, abs/1707.05589, 2017. URL http://arxiv.org/abs/1707.05589.

Gábor Melis, Tomáš Kočiskỳ, and Phil Blunsom. Mogrifier LSTM. *International Conference on Learning Representations*, 2020.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 1045–1048, 2010.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

John W. Mohr and Petko Bogdanov. Topic models: What they are and why they matter, 2013.

Joe O'Connor and Jacob Andreas. What context features can transformer language models use? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 851–864, August 2021. doi: 10.18653/v1/2021.acl-long.70. URL https://aclanthology.org/2021.acl-long.70.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*, 2020.

Mehdi Rezaee and Francis Ferraro. A discrete variational recurrent topic model without the reparametrization trick. *Advances in Neural Information Processing Systems*, 33:13831–13843, 2020.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. Do neural dialog systems use the conversation history effectively? An empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 32–37, July 2019. doi: 10.18653/v1/P19-1004. URL https://aclanthology.org/P19-1004.

Hongyin Tang, Miao Li, and Beihong Jin. A topic augmented text generation model: Joint learning of semantics and structural features. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 5090–5099, November 2019. doi: 10.18653/v1/D19-1513. URL https://aclanthology.org/D19-1513.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Martin J. Wainwright, Michael I. Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

Hanna Wallach, David Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 22, 2009. URL https://proceedings.neurips.cc/paper/2009/file/0d0871f0806eae32d30983b62252da50-Paper.pdf.

Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. Topic compositional neural language model. In *International Conference on Artificial Intelligence and Statistics*, pp. 356–365, 2018.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 166–177, June 2019. doi: 10.18653/v1/N19-1015. URL https://aclanthology.org/N19-1015.

Ronald J. Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2(4):490–501, 1990.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. *CoRR*, abs/1711.03953, 2017. URL http://arxiv.org/abs/1711.03953.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization, 2014. URL https://arxiv.org/abs/1409.2329.

A LSTM

The LSTM (Hochreiter & Schmidhuber, 1997) components are

$$
\begin{aligned}
i_t &= \sigma(\mathbf{W}_i v_t + \mathbf{U}_i h_{t-1} + b_i) \\
f_t &= \sigma(\mathbf{W}_f v_t + \mathbf{U}_f h_{t-1} + b_f) \\
o_t &= \sigma(\mathbf{W}_o v_t + \mathbf{U}_o h_{t-1} + b_i) \\
\hat{c}_t &= \tanh(\mathbf{W}_c v_t + \mathbf{U}_c h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \hat{c}_t \\
h_t &= o_t \odot \tanh(c_t).
\end{aligned}
$$

The symbol $\odot$ denotes element-wise product, while i_t, f_t, o_t are the input, forget, and output activations at time t. Additionally, v_t, h_t, c_t are the input word embedding, hidden state, and cell state at time t. Finally, $\mathbf{W}, \mathbf{U}, b$ are model parameters.

Table 4: Dataset statistics.

Dataset	Vocab Size	Training		Validation		Test	
		Docs	Tokens	Docs	Tokens	Docs	Tokens
APNEWS	34231	50K	15M	2K	0.6M	2K	0.6M
IMDB	36009	75K	20M	12.5K	0.3M	12.5K	0.3M
BNC	43703	15K	18M	1K	1M	1K	1M

B GRU

The GRU (Cho et al., 2014) components are

$$
\begin{aligned}
z_t &= \sigma(\mathbf{W}_z v_t + \mathbf{U}_z h_t + b_z) \\
r_t &= \sigma(\mathbf{W}_r v_t + \mathbf{U}_r h_t + b_r) \\
\hat{h}_t &= \tanh(\mathbf{W}_h s + \mathbf{U}_h (r_t \odot h_t) + b_h) \\
h'_t &= (1 - z_t) \odot h_t + z_t \odot \hat{h}_t.
\end{aligned}
$$

Here, z_t and r_t are the update and reset gate activations at time t. Meanwhile, v_t and h_t are the input vector and the hidden state at time t, while $\mathbf{W}, \mathbf{U}$, and b are model parameters.

C Datasets

We evaluate on three publicly available corpora. 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). BNC is the written portion of the British National Corpus (Consortium, 2007), which contains excerpts from journals, books, letters, essays, memoranda, news, and other types of text. A random subset of APNEWS and BNC is selected for the experiments.

The data is preprocessed as follows. Documents are lowercased and tokenized using Stanford CoreNLP (Klein & Manning, 2003). Tokens that occur less than 10 times are replaced with the UNK token. The SOS token is prepended to the start of each document; the EOS token is appended to the end of each document. While we use the same vocabulary and tokenization as Lau et al. (2017), we do not add extra SOS and EOS tokens to the beginning and end of each sentence, so our perplexity numbers are not directly comparable to Lau et al. (2017); Rezaee & Ferraro (2020); Guo et al. (2020), who evaluate on the same datasets. When we redo the reproducibility experiment to use their original preprocessing settings, the trends in model performance are nearly identical to the results in this paper. Table 4 shows the dataset statistics. We use the same training, validation, and test splits as Lau et al. (2017).

Each model uses the same vocabulary for next-word prediction, so predictive performance is comparable across models. Models have different specifications for the number of words in the topic model component. For rGBN-RNN and TDLM, we follow the vocabulary preprocessing steps outlined in the respective papers. We exclude the top 0.1% most frequent tokens and tokens that appear in the Mallet stop word list. For TopicRNN and VRTM, we additionally exclude words that appear in less than 100 documents, following Wang et al. (2018).

D Experiment Settings

We train all models on single GPUs with a language model batch size of 64. LSTM-LM, TopicRNN, VRTM, and TDLM are implemented in our codebase in Pytorch 1.13.1. We use the original implementation of rGBN-RNN, which uses Tensorflow 1.9.

We note differences between our experiment settings and the original papers here; all other settings are the same as in the original papers, and we refer the reader to them for details.

Table 5: Randomly selected learned topics from each model on APNEWS.

Model	Topics
TopicRNN	athens banks atlanta georgia newark lenders egypt foreclosure reagan castle considerable insights pioneer mounted suggestion authorize dividend prevents shelves hurdle harmful five-year rouge discharged inch felonies yorkers soaring height discover
TDLM	airline pilots tsa passengers faa airlines va plane flight cartel shooting shot officers killed murder death car suspect wounded gun museum artifacts exhibit exhibition smithsonian paintings exhibits art museums sculpture
rGBN-RNN	market economy fed japan investors index markets financial banks china dogs dog family animal find called killed shelter hansen living gas primary election environmental race murkowski congressional groups process running
VRTM	counties percentage billion believes line cent albany reported center caliber counties percentage billion center reported line park believes caliber family counties percentage billion center reported line caliber believes check albany

For TopicRNN, we increase the model size to be comparable to the other models. Specifically, for the LSTM, we use 600 hidden units instead of 300. To predict the stop word indicators, we use a 2-layer feedforward network before the final linear layer to be consistent with VRTM. For VRTM, we use a backpropagation sequence length of 30 instead of 45 to be consistent with the other models. We also initialize the embeddings using the same pretrained embeddings as the other models, as we found this to help VRTM's performance. For TDLM, we use a batch size of 320 instead of 64 for the topic model batches in order to speed up training. As noted in the main paper, for rGBN-RNN, we use a model size of 600-600-600 because their public code only supports 3-layer models with same-size RNN layers.

As described in the main text, each model in Table 1 is trained until the validation loss does not improve for 10 epochs. After convergence, we use the checkpoint with the best validation perplexity. For each model, we perform three runs with random initializations all trained until convergence. We report the mean of these runs along with their standard deviations.

For LDA, we use the Mallet implementation, which performs inference using Gibbs sampling. The hyperparameters are: α (topic density) = 50, β (word density) = 0.01, # iterations = 1000.

E Topic-Guided LM Topics

Table 5 includes randomly sampled topics from each topic-guided language model fit to APNEWS.