A Discrete Variational Recurrent Topic Model without the Reparametrization Trick

Mehdi Rezaee
Department of Computer Science
University of Maryland Baltimore County
Baltimore, MD 21250 USA
rezaee1@umbc.edu

Francis Ferraro
Department of Computer Science
University of Maryland Baltimore County
Baltimore, MD 21250 USA
ferraro@umbc.edu

Abstract

We show how to learn a neural topic model with discrete random variables—one that explicitly models each word's assigned topic—using neural variational inference that does not rely on stochastic backpropagation to handle the discrete variables. The model we utilize combines the expressive power of neural methods for representing sequences of text with the topic model's ability to capture global, thematic coherence. Using neural variational inference, we show improved perplexity and document understanding across multiple corpora. We examine the effect of prior parameters both on the model and variational parameters, and demonstrate how our approach can compete and surpass a popular topic model implementation on an automatic measure of topic quality.

1 Introduction

With the successes of deep learning models, neural variational inference (NVI) [27]—also called variational autoencoders (VAE)—has emerged as an important tool for neural-based, probabilistic modeling [19, 40, 37]. NVI is relatively straight-forward when dealing with continuous random variables, but necessitates more complicated approaches for discrete random variables.

The above can pose problems when we use discrete variables to model data, such as capturing both syntactic and semantic/thematic word dynamics in natural language processing (NLP). Short-term memory architectures have enabled Recurrent Neural Networks (RNNs) to capture local, syntactically-driven lexical dependencies, but they can still struggle to capture longer-range, thematic dependencies [44, 29, 5, 20]. Topic modeling, with its ability to effectively cluster words into thematically-similar groups, has a rich history in NLP and semantics-oriented applications [4, 3, 50, 6, 46, 35, i.a.]. However, they can struggle to capture shorter-range dependencies among the words in a document [8]. This suggests these problems are naturally complementary [42, 53, 45, 41, 27, 15, 8, 48, 51, 22].

NVI has allowed the above recurrent topic modeling approaches to be studied, but with two primary modifications: the discrete variables can be reparametrized and then sampled, or each word's topic assignment can be analytically marginalized out, *prior* to performing any learning. However, previous work has shown that topic models that preserve explicit topics yield higher quality topics than similar models that do not [25], and recent work has shown that topic models that have relatively consistent word-level topic assignments are preferred by end-users [24]. Together, these suggest that there are benefits to preserving these assignments that are absent from standard RNN-based language models. Specifically, preservation of word-level topics within a recurrent neural language model may both improve language prediction as well as yield higher quality topics.

To illustrate this idea of thematic vs. syntactic importance, consider the sentence "*She received bachelor's and master's degrees in electrical engineering from Anytown University*." While a topic

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

model can easily learn to group these "education" words together, an RNN is designed explicitly to capture the sequential dependencies, and thereby predict coordination (*"and"*), metaphorical (*"in"*) and transactional (*"from"*) concepts given the previous *thematically*-driven tokens.

In this paper we reconsider core modeling decisions made by a previous recurrent topic model [8], and demonstrate how the discrete topic assignments can be maintained in both learning and inference without resorting to reparametrizing them. In this reconsideration, we present a simple yet efficient mechanism to learn the dynamics between thematic and non-thematic (e.g., syntactic) words. We also argue the design of the model's priors still provides a key tool for language understanding, even when using neural methods. The main contributions of this work are:

1. We provide a recurrent topic model and NVI algorithm that (a) explicitly considers the surface dynamics of modeling with thematic and non-thematic words and (b) maintains word-level, discrete topic assignments without relying on reparametrizing or otherwise approximating them.
2. We analyze this model, both theoretically and empirically, to understand the benefits the above modeling decisions yield. This yields a deeper understanding of certain limiting behavior of this model and inference algorithm, especially as it relates to past efforts.
3. We show that careful consideration of priors and the probabilistic model still matter when using neural methods, as they can provide greater control over the learned values/distributions. Specifically, we find that a Dirichlet prior for a document's topic proportions provides fine-grained control over the statistical structure of the learned variational parameters vs. using a Gaussian distribution.

Our code, scripts, and models are available at `https://github.com/mmrezaee/VRTM`.

2 Background

Latent Dirichlet Allocation [4, LDA] defines an admixture model over the words in documents. Each word $w_{d,t}$ in a document d is stochastically drawn from one of K topics—discrete distributions over $\mathcal{V}$ vocabulary words. We use the discrete variable $z_{d,t}$ to represent the topic that the tth word is generated from. We formally define the well-known generative story as drawing document-topic proportions $\theta_d \sim \mathrm{Dir}(\alpha)$, then drawing individual topic-word assignments $z_{d,t} \sim \mathrm{Cat}(\theta_d)$, and finally generating each word $w_{d,t} \sim \mathrm{Cat}(\beta_{z_t})$, based on the topics ($\beta_{k,v}$ is the conditional probability of word v given topic k).

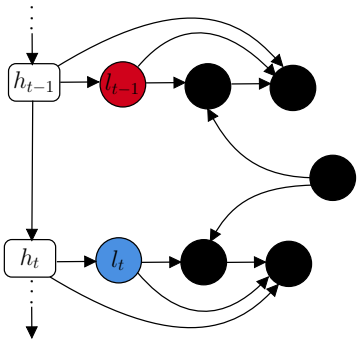
Two common ways of learning an LDA model are either through Monte Carlo sampling techniques, that iteratively sample states for the latent random variables in the model, or variational/EM-based methods, which minimize the distribution distance between the posterior $p(\theta, z|w)$ and an approximation $q(\theta, z; \gamma, \phi)$ to that posterior that is controlled by learnable parameters γ and ϕ. Commonly, this means minimizing the negative KL divergence, $-\mathrm{KL}(q(\theta, z; \gamma, \phi) \| p(\theta, z|w))$ with respect to γ and ϕ. We focus on variational inference as it cleanly allows neural components to be used.

Supported by work that has studied VAEs for a variety of distributions [12, 33], the use of deep learning models in the VAE framework for topic modeling has received recent attention [42, 53, 45, 41, 27, 15]. This complements other work that focused on extending the concept of topic modeling using undirected graphical models [43, 17, 21, 31].

Traditionally, stop words have been a nuisance when trying to learn topic models. Many existing algorithms ignore stop-words in initial pre-processing steps, though there have been efforts that try to alleviate this problem. Wallach et al. [47] suggested using an asymmetric Dirichlet prior distribution to include stop-words in some isolated topics. Eisenstein et al. [9] introduced a constant background factor derived from log-frequency of words to avoid the need for latent switching, and Paul [36] studied structured residual models that took into consideration the use of stop words. These approaches do not address syntactic language modeling concerns.

A recurrent neural network (RNN) can address those concerns. A basic RNN cell aims to predict each word w_t given previously seen words $w_{<t} = \{w_0, w_1, \ldots, w_{t-1}\}$. Words are represented with embedding vectors and the model iteratively computes a new representation $h_t = f(w_{<t}, h_{t-1})$, where f is generally a non-linear function.

There have been a number of attempts at leveraging the combination of RNNs and topic models to capture local and global semantic dependencies [8, 48, 51, 22, 39]). Dieng et al. [8] passed the



(a) Graphical structure underlying the VRTM model. The shaded gray circles depict observed words. red nodes denotes non-thematic indicators ($l_t = 0$) and blue nodes show a thematic indicators ($l_t = 1$).

1. Draw a document topic vector $\theta \sim \text{Dir}(\alpha)$.
2. For each of the T observed words ($t = 1 \dots T$):
   - Compute the recurrent representation $h_t = f(x_t, h_{t-1})$, where $x_t = w_{t-1}$.
   - Draw $l_t \sim \text{Bern}\,(\rho)$, indicating whether word t is thematically relevant.
   - Draw a topic $z_t \sim p(z_t|l_t, \theta)$.
   - Draw the word $w_t \sim p(w_t|z_t, l_t; h_t, \beta)$ using eq. 2.

(b) The generative story of the model we study. The story largely follows Dieng et al. [8], though we redefine the prior on θ and the decoder $p(w_t|z_t, l_t; h_t, \beta)$. Beyond these, the core differences lie in how we approach learning the model.

Figure 1: An unrolled graphical model (Fig. 1a) for the corresponding generative story (Fig. 1b). Tokens are fed to a recurrent neural network (RNN) cell, which computes a word-specific representation h_t. With this representation, the thematic word indicator l_t, and inferred topic proportions θ shared across the document, we assign the topics z_t, which allows us to generate each word w_t. Based on our decoding model (eq. 2), the thematic indicators l_t topic assignments help trade-off between the recurrent representations and the (stochastically) assigned topics z_t.

topics to the output of RNN cells through an additive procedure while Lau et al. [22] proposed using convolutional filters to generate document-level features and then exploited the obtained topic vectors in a gating unit. Wang et al. [48] provided an analysis of incorporating topic information inside a Long Short-Term Memory (LSTM) cell, followed by a topic diversity regularizer to make a complete compositional neural language model. Nallapati et al. [34] introduced sentence-level topics as a way to better guide text generation: from a document's topic proportions, a topic for a sentence would be selected and then used to generate the words in the sentence. The generated topics are shared among the words in a sentence and conditioned on the assigned topic. Li et al. [23] followed the same strategy, but modeled a document as a sequence of sentences, such that recurrent attention windows capture dependencies between successive sentences. Wang et al. [49] recently proposed using a normal distribution for document-level topics and a Gaussian mixture model (GMM) for word topics assignments. While they incorporate both the information from the previous words and their topics to generate a sentence, their model is based on a mixture of experts, with word-level topics neglected. To cover both short and long sentences, Gupta et al. [14] introduced an architecture that combines the LSTM hidden layers and the pre-trained word embeddings with a neural auto-regressive topic model. One difference between our model and theirs is that their focus is on predicting the words within a context, while our aim is to generate words given a particular topic without marginalizing.

3 Method

We first formalize the problem and propose our variationally-learned recurrent neural topic model (**VRTM**) that accounts for each word's generating topic. We discuss how we learn this model and the theoretical benefits of our approach. Subsequently we demonstrate the empirical benefits of our approach (Sec. 4).

3.1 Generative Model

Fig. 1 provides both an unrolled graphical model and the full generative story. We define each document as a sequence of T words $\mathbf{w}_d = \{w_{d,t}\}_{t=1}^{T}$. For simplicity we omit the document index d when possible. Each word has a corresponding generating topic z_t, drawn from the document's topic proportion distribution θ. As in standard LDA, we assume there are K topics, each represented as a $\mathcal{V}$-dimensional vector β_k where $\beta_{k,v}$ is the probability of word v given topic k. However, we sequentially model each word via an RNN. The RNN computes h_t for each token, where h_t is designed to consider the previous $t-1$ words observed in sequence. During training, we define a

sequence of observed semantic indicators (one per word) $\mathbf{l} = \{l_t\}_{t=1}^T$, where $l_t = 1$ if that token is thematic (and 0 if not); a sequence of latent topic assignments $\mathbf{z} = \{z_t\}_{t=1}^T$; and a sequence of computed RNN states $\mathbf{h} = \{h_t\}_{t=1}^T$. We draw each word's topic from an assignment distribution $p(z_t|l_t, \theta)$, and generate each word from a decoding/generating distribution $p(w_t|z_t, l_t; h_t, \beta)$.

The joint model can be decomposed as

$$p(\mathbf{w}, \mathbf{l}, \mathbf{z}, \theta; \beta, \mathbf{h}) = p(\theta; \alpha) \prod_{t=1}^{T} p(w_t|z_t, l_t; h_t, \beta) p(z_t|l_t, \theta) p(l_t; h_t), \tag{1}$$

where α is a vector of positive parameters governing what topics are likely in a document (θ). The corresponding graphical model is shown in Fig. 1a. Our decoder is

$$p(w_t = v|z_t = k; h_t, l_t, \beta) \propto \exp\left(p_v^\top h_t + l_t \beta_{k,v}\right), \tag{2}$$

where p_v stands for the learned projection vector from the hidden space to output. When the model faces a non-thematic word it just uses the output of the RNN and otherwise it uses a mixture of LDA and RNN predictions. As will be discussed in Sec. 3.2.2, separating h_t and $z_t = k$ in this way allows us to marginalize out z_t during inference without reparametrizing it.

A similar type of decoder has been applied by both Dieng et al. [8] and Wen and Luong [51] to combine the outputs of the RNN and LDA. However, as they marginalize out the topic assignments their decoders do not condition on the topic assignment z_t. Mathematically, their decoders compute $\beta_v^\top \theta$ instead of $\beta_{k,v}$, and compute $p(w_t = v|\theta; h_t, l_t, \beta) \propto \exp\left(p_v^\top h_t + l_t \beta_v^\intercal \theta\right)$. This new decoder structure helps us preserve the word-level topic information which is neglected in other models. This seemingly small change has out-sized empirical benefits (Tables 1b and 2).[1]

Note that we explicitly allow the model to trade off between thematic and non-thematic words. This is very much similar to Dieng et al. [8]. We assume during training these indicators are observable, though their occurrence is controlled via the output of a neural factor: $\rho = \sigma\big(g(h_t)\big)$ (σ is the sigmoid function). Using the recurrent network's representations, we allow the model to learn the presence of thematic words to be learned via—a long recognized area of interest [11, 26, 52, 16].

For a K-topic VRTM, we formalize this intuition by defining $p(z_t = k|l_t, \theta) = \frac{1}{K}$, if $l_t = 0$ and θ_k otherwise. The rationale behind this assumption is that in our model the RNN is sufficient to draw non-thematic words and considering a particular topic for these words makes the model unnecessarily complicated. Maximizing the data likelihood $p(\mathbf{w}, \mathbf{l})$ in this setting is intractable due to the integral over θ. Thus, we rely on amortized variational inference as an alternative approach. While classical approaches to variational inference have relied on statistical conjugacy and other approximations to make inference efficiently computable, the use of neural factors complicates such strategies. A notable difference between our model and previous methods is that we take care to account for each word and its assignment in both our model and variational approximation.

3.2 Neural Variational Inference

We use a mean field assumption to define the variational distribution $q(\theta, \mathbf{z}|\mathbf{w}, \mathbf{l})$ as

$$q(\theta, \mathbf{z}|\mathbf{w}, \mathbf{l}) = q(\theta|\mathbf{w}, \mathbf{l}; \gamma) \prod_{t=1}^{T} q(z_t|w_t, l_t; \phi_t), \tag{3}$$

where $q(\theta|\mathbf{w}, \mathbf{l}, \gamma)$ is a Dirichlet distribution parameterized by γ and we define $q(z_t)$ similarly to $p(z_t)$: $q(z_t = k|w_t, l_t; \phi_t) = \frac{1}{K}$ if $l_t = 0$ and the neural parametrized output ϕ_t^k otherwise. In Sec. 3.2.1 we define γ and ϕ_t in terms of the output of neural factors. These allow us to "encode" the input into our latent variables, and then "decode" them, or "reconstruct" what we observe.

Variational inference optimizes the evidence lower bound (ELBO). For a single document (subscript omitted for clarity), we write the ELBO as:

$$\mathcal{L} = \mathbb{E}_{q(\theta, \mathbf{z}|\mathbf{w}, \mathbf{l})} \left[\sum_{t=1}^{T} \log p(w_t|z_t, l_t; h_t, \beta) + \log \frac{p(z_t|l_t, \theta)}{q(z_t|w_t, l_t)} + \log p(l_t; h_t) + \log \frac{p(\theta)}{q(\theta|\mathbf{w}, \mathbf{l})} \right]. \tag{4}$$

[1] Since the model drives learning and inference, this model's similarities and adherence to both foundational [4] and recent neural topic models [8] are intentional. We argue, and show empirically, that this adherence yields language modeling performance and allows more exact marginalization during learning.

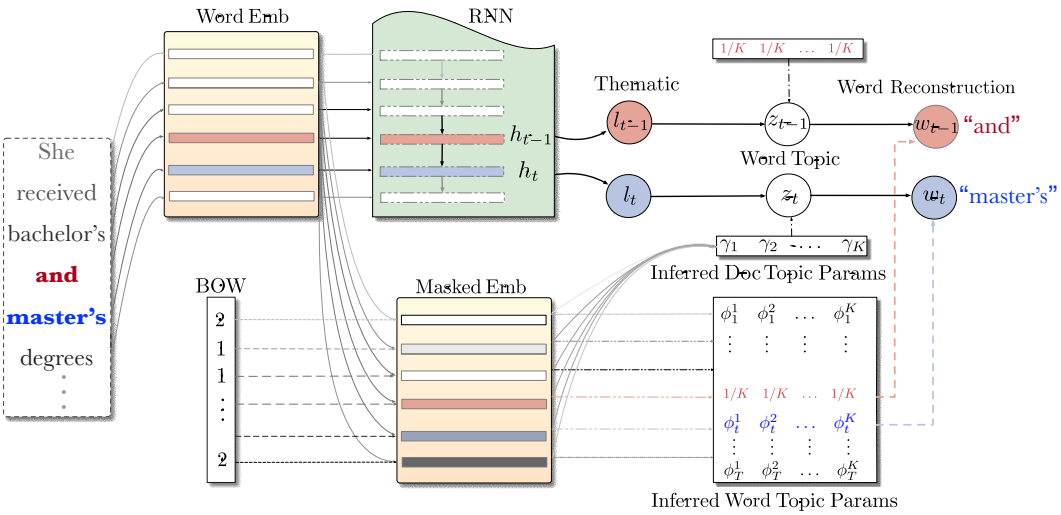


Figure 2: Our encoder and decoder architectures. Word embeddings, learned from scratch, are both provided to a left-to-right recurrent network and used to create bag-of-words masked embeddings (masking non-thematic words, i.e., $l_t = 0$, by 0). These masked embeddings are used to compute γ and ϕ—the variational parameters—while the recurrent component computes h_t (which is used to compute the thematic predictor and reconstructed words, as shown in Fig. 1). For clarity, we have omitted the connection from h_t to w_t (these are shown in Fig. 1a).

This objective can be decomposed into separate loss functions and written as $\mathcal{L} = \mathcal{L}_w + \mathcal{L}_z + \mathcal{L}_\phi + \mathcal{L}_l + \mathcal{L}_\theta$, where, using $\langle\cdot\rangle = \mathbb{E}_{q(\theta,\mathbf{z}|\mathbf{w},\mathbf{l})}[\cdot]$, we have $\mathcal{L}_w = \langle\sum_t \log p(w_t|z_t, l_t; h_t, \beta)\rangle$, $\mathcal{L}_z = \langle\sum_t \log p(z_t|l_t, \theta)\rangle$, $\mathcal{L}_\phi = \langle\sum_t \log q(z_t|w_t, l_t)\rangle$, $\mathcal{L}_l = \langle\sum_t \log p(l_t; h_t)\rangle$, and $\mathcal{L}_\theta = \langle\log \frac{p(\theta)}{q(\theta|\mathbf{w},\mathbf{l})}\rangle$.

The algorithmic complexity, both time and space, will depend on the exact neural cells used. The forward pass complexity for each of the separate summands follows classic variational LDA complexity.

In Fig. 2 we provide a system overview of our approach, which can broadly be viewed as an encoder (the word embedding, bag-of-words, masked word embeddings, and RNN components) and a decoder (everything else). Within this structure, our aim is to maximize $\mathcal{L}$ to find the best variational and model parameters. To shed light onto what the ELBO is showing, we describe each term below.

3.2.1 Encoder

Our encoder structure maps the documents into variational parameters. Previous methods posit that while an RNN can adequately perform language modeling (LM) using word embeddings for all vocabulary words, a modified, bag-of-words-based embedding representation is an effective approach for capturing thematic associations [8, 48, 22]. We note however that more complex embedding methods can be studied in the future [38, 7, 2].

To avoid "representation degeneration" [13], care must be taken to preserve the implicit semantic relationships captured in word embeddings.[2] Let $\mathcal{V}$ be the overall vocabulary size. For RNN language modeling [LM], we represent each word v in the vocabulary with an E-dimensional real-valued vector $e_v \in \mathbb{R}^E$. As a result of our model definition, the topic modeling [TM] component of VRTM effectively operates with a reduced vocabulary; let $\mathcal{V}'$ be the vocabulary size excluding non-thematic words ($\mathcal{V}' \leq \mathcal{V}$). Each document is represented by a $\mathcal{V}'$-dimensional integer vector $\mathbf{c} = \langle c_1, c_2, \cdots, c_{\mathcal{V}'}\rangle$, recording how often each thematic word appeared in the document.

In contrast to a leading approach [49], we use the entire e_v vocabulary embeddings for LM and mask out the non-thematic words to compute the variational parameters for TM as $e_v^{\text{TM}} = e_v \odot c_v^{\text{TM}}$, where $\odot$ denotes element-wise multiplication and $c_v^{\text{TM}} = c_v$ if v is thematic (0 otherwise). We refer to e_v^{TM} as the masked embedding representation. This motivates the model to learn the word meaning

[2] "Representation degeneration" happens when after training the model, the word embeddings are not well separated in the embedding space and all of them are highly correlated [13].

and importance of word counts jointly. The token representations are gathered into $e_w^{\text{TM}} \in \mathbb{R}^{T \times E}$, which is fed to a feedforward network with one hidden layer to infer word-topics ϕ. Similarly, we use tensordot to produce the γ parameters for each document defined as $\boldsymbol{\gamma} = \text{softplus}(e_w^{\text{TM}} \otimes W_\gamma)$, where $W_\gamma \in \mathbb{R}^{T \times E \times K}$ is the weight tensor and the summation is over the common axes.

3.2.2 Decoder/Reconstruction Terms

The first term in the objective, $\mathcal{L}_w$, is the reconstruction part, which simplifies to

$$\mathcal{L}_w = \mathbb{E}_{q(\theta,\mathbf{z}|\mathbf{w},\mathbf{l})}\left[\sum_{t=1}^{T} \log p(w_t|z_t, l_t; h_t, \beta)\right] = \sum_{t=1}^{T}\left[\mathbb{E}_{q(z_t|w_t,l_t;\phi_t)} \log p(w_t|z_t, l_t; h_t, \beta)\right]. \quad (5)$$

For each word w_t, if $l_t = 0$, then the word is not thematically relevant. Because of this, we just use h_t. Otherwise ($l_t = 1$) we let the model benefit from both the globally semantically coherent topics β and the locally fluent recurrent states h_t. With our definition of $p(z_t|\theta)$, $\mathcal{L}_w$ simplifies to $\mathcal{L}_w = \sum_{t:l_t=1} \sum_{k=1}^{K} \phi_t^k \log p(w_t|z_t; h_t, \beta) + \sum_{t:l_t=0} \log p(w_t; h_t)$. Finally, note that there is an additive separation between h_t and $l_t \beta_{z_t,v}$ in the word reconstruction model (eq. 2). The marginalization that occurs per-word can be computed as $p_v^\intercal h_t + l_t \langle \beta_{z_t,v} \rangle_{z_t} - \langle \log Z(z_t, l_t; h_t, \beta) \rangle_{z_t}$, where $Z(z_t, l_t; h_t, \beta)$ represents the per-word normalization factor. This means that the discrete z can be marginalized out and do not need to be sampled or approximated.

Given the general encoding architecture used to compute $q(\theta)$, computing $\mathcal{L}_z = \mathbb{E}_{q(\theta,\mathbf{z}|\mathbf{w},\mathbf{l})}\left[\sum_{t=1}^{T} \log p(z_t|l_t, \theta)\right]$ requires computing two expectations. We can exactly compute the expectation over z, but we approximate the expectation over θ with S Monte Carlo samples from the learned variational approximation, with $\theta^{(s)} \sim q(\theta)$. Imposing the basic assumptions from $p(z)$ and $q(z)$, we have $\mathcal{L}_z \approx \frac{1}{S} \sum_{s=1}^{S} \sum_{t:l_t=1} \sum_{k=1}^{K} \phi_t^k \log \theta_k^{(s)} - C \log K$, where C is the number of the non-thematic words in the document and $\theta_k^{(j)}$ is the output of Monte Carlo sampling. Similarly, we have $\mathcal{L}_\phi = -\mathbb{E}_{q(\mathbf{z}|\mathbf{w},\mathbf{l};\phi)}\left[\log q(\mathbf{z}|\mathbf{w}, \mathbf{l}; \phi)\right] = -\sum_{t:l_t=1} \sum_{k=1}^{K} \phi_t^k \log \phi_t^k + C \log K$.

Modeling which words are thematic or not can be seen as a negative binary cross-entropy loss: $\mathcal{L}_l = \mathbb{E}_{q(\theta,\mathbf{z}|\mathbf{w},\mathbf{l})}\left[\log p(l_t; h_t)\right] = \sum_{t=1}^{T} \log p(l_t; h_t)$. This term is a classifier for l_t that motivates the output of the RNN to discriminate between thematic and non-thematic words. Finally, as a KL-divergence between distributions of the same type, $\mathcal{L}_\theta$ can be calculated in closed form [18].

We conclude with a theorem, which shows VRTM is an extension of RNN under specific assumptions. This demonstrates the flexibility of our proposed model and demonstrates why, in addition to natural language processing considerations, the renewed accounting mechanisms for thematic vs. non-thematic words in both the model *and* inference is a key idea and contribution. For the proof, please see the appendix. The central idea is to show that all the terms related to TM either vanish or are constants; it follows intuitively from the ELBO and eq. 2.

Theorem 1. *If in the generative story $\rho = 0$, and we have a uniform distribution for the prior and variational topic posterior approximation $\big(p(\theta) = q(\theta|\mathbf{w}, \mathbf{l}) = 1/K\big)$, then VRTM reduces to a Recurrent Neural Network to just reconstruct the words given previous words.*

4 Experimental Results

Datasets We test the performance of our algorithm on the APNEWS, IMDB and BNC datasets that are publicly available.[3] Roughly, there are between 7.7k and 9.8k vocab words in each corpus, with between 15M and 20M training tokens each; Table A1 in the appendix details the statistics of these datasets. APNEWS contains 54k newswire articles, IMDB contains 100k movie reviews, and BNC contains 17k assorted texts, such as journals, books excerpts, and newswire. These are the same datasets including the train, validation and test splits, as used by prior work, where additional details can be found [48]. We use the publicly provided tokenization and following past work we lowercase all text and map infrequent words (those in the bottom 0.01% of frequency) to a special <unk> token. Following previous work [8], to avoid overfitting on the BNC dataset we grouped 10 documents in the training set into a new pseudo-document.

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

Setup Following Dieng et al. [8], we find that, for basic language modeling and core topic modeling, identifying which words are/are not stopwords is a sufficient indicator of thematic relevance. We define $l_t = 0$ if w_t is within a standard English stopword list, and 1 otherwise.[4] We use the softplus function as the activation function and batch normalization is employed to normalize the inputs of variational parameters. We use a single-layer recurrent cell; while some may be surprised at the relative straightforwardness of this choice, we argue that this choice is a benefit. Our models are intentionally trained using *only* the provided text in the corpora so as to remain comparable with previous work and limit confounding factors; we are not using pre-trained models. While transformer methods are powerful, there are good reasons to research non-transformer approaches, e.g., the data & computational requirements of transformers. We demonstrate the benefits of separating the "semantic" vs. "syntactic" aspects of LM, and could provide the blueprint for future work into transformer-based TM. For additional implementation details, please see appendix C.

Language Modeling Baselines In our experiments, we compare against the following methods:

- **basic-LSTM**: A single-layer LSTM with the same number of units like our method implementation but without any topic modeling, i.e. $l_t = 0$ for all tokens.
- **LDA+LSTM** [22]: Topics, from a trained LDA model, are extracted for each word and concatenated with the output of an LSTM to predict next words.
- **Topic-RNN** [8]: This work is the most similar to ours. They use the similar decoder strategy (Eq. 2) but marginalize topic assignments prior to learning.
- **TDLM** [22]: A convolutional filter is applied over the word embeddings to capture long range dependencies in a document and the extracted features are fed to the LSTM gates in the reconstruction phase.
- **TCNLM** [48]: A Gaussian vector defines document topics, then a set of expert networks are defined inside an LSTM cell to cover the language model. The distance between topics is maximized as a regularizer to have diverse topics.
- **TGVAE** [49]: The structure is like TCNLM, but with a Gaussian mixture model to define each sentence as a mixture of topics. The model inference is done by using K householder flows to increase the flexibility of variational distributions.

4.1 Evaluations and Analysis

Perplexity For an RNN, we used a single-layer LSTM with 600 units in the hidden layer, set the size of embedding to be 400, and had a fixed/maximum sequence length of 45. We also present experiments demonstrating the performance characteristics of using basic RNN, GRU and LSTM cells in Table 1a. Note that although our embedding size is higher we do not use pretrained word embeddings, but instead learn them from scratch via end-to-end training.[5] The perplexity values of the baselines and our VRTM across our three heldout evaluation sets are shown in Table 1b. We note that TGVAE [49] also compared against, and outperformed, all the baselines in this work. In surpassing TGVAE, we are also surpassing them. We find that our proposed method achieves lower perplexity than LSTM. This is consistent with Theorem 1. We observe that increasing the number of topics yields better overall perplexity scores. Moreover, as we see VRTM outperforms other baselines across all the benchmark datasets.

Topic Switch Percent Automatically evaluating the quality of learned topics is notoriously difficult. While "coherence" [30] has been a popular automated metric, it can have peculiar failure points especially regarding very common words [36]. To counter this, Lund et al. [24] recently introduced switch percent (SwitchP). SwitchP makes the very intuitive yet simple assumption that "good" topics will exhibit a type of inertia: one would not expect adjacent words to use many different topics. It aims to show the consistency of the adjacent word-level topics by computing the number of times the same topic was used between adjacent words: $(T_d - 1)^{-1} \sum_{t=1}^{T_d - 1} \delta(z_t, z_{t+1})$, where $z_t = \arg\max_k \{\phi_t^1, \phi_t^2, \ldots, \phi_t^K\}$, and T_d is the length of document after removing the stop-words.

[4] Unlike Asymmetric Latent Dirichlet Allocation (ALDA) [47], which assigns a specific topic for stop words, we assume that stop words *do not* belong to any specific topic.

[5] Perplexity degraded slightly with pretrained embeddings, e.g., with $T = 300$, VRTM-LSTM on APNEWS went from 51.35 (from scratch) to 52.31 (word2vec).

Model	APNEWS		IMDB	
	300	**400**	**300**	**400**
RNN ($T = 10$)	66.21	61.48	69.48	79.94
RNN ($T = 30$)	60.03	61.21	80.65	76.04
RNN ($T = 50$)	59.24	60.64	66.45	66.42
GRU ($T = 10$)	61.57	58.82	75.21	67.31
GRU ($T = 30$)	63.40	58.59	84.27	67.61
GRU ($T = 50$)	59.09	57.91	84.45	63.59
LSTM ($T = 10$)	55.19	54.31	60.14	59.82
LSTM ($T = 30$)	53.76	51.47	58.27	54.36
LSTM ($T = 50$)	51.35	47.78	57.70	51.08

(a) Test perplexity for different RNN cells and embedding sizes (300 vs. 400). T denotes the number of topics.

Methods	APNEWS	IMDB	BNC
basic-LSTM	62.79	70.38	100.07
LDA+LSTM ($T = 50$)	57.05	69.58	96.42
Topic-RNN ($T = 50$)	56.77	68.74	94.66
TDLM ($T = 50$)	53.00	63.67	91.42
TCNLM ($T = 50$)	52.75	63.98	87.98
TGVAE ($T = 10$)	$\leq$55.77	$\leq$62.22	$\leq$91.19
TGVAE ($T = 30$)	$\leq$51.27	$\leq$59.45	$\leq$88.34
TGVAE ($T = 50$)	$\leq$48.73	$\leq$57.11	$\leq$87.86
VRTM-LSTM ($T = 10$)	$\leq$54.31	$\leq$59.82	$\leq$92.89
VRTM-LSTM ($T = 30$)	$\leq$51.47	$\leq$54.36	$\leq$89.26
VRTM-LSTM ($T = 50$)	$\leq$ **47.78**	$\leq$ **51.08**	$\leq$ **86.33**

(b) Test perplexity, as reported in previous works.[6] T denotes the number of topics. Consistent with Wang et al. [49] we report the maximum of three VRTM runs.

Table 1: Test set perplexity (lower is better) of VRTM demonstrates the effectiveness of our approach at learning a topic-based language model. In 1a we demonstrate the stability of VRTM using different recurrent cells. In 1b, we demonstrate our VRTM-LSTM model outperforms prior neural topic models. We do not use pretrained word embeddings.

Topics	APNEWS		IMDB		BNC	
	LDA	VRTM	LDA	VRTM	LDA	VRTM
5	0.26	0.59	0.24	0.52	0.24	0.51
10	0.18	0.43	0.14	0.35	0.15	0.40
15	0.14	0.33	0.12	0.31	0.13	0.35
30	0.10	0.31	0.09	0.28	0.10	0.23
50	0.08	0.20	0.07	0.26	0.07	0.20

(a) SwitchP (higher is better) for VRTM vs LDA VB [4] averaged across three runs. Comparisons are valid within a corpus and for the same number of topics.

Topics	APNEWS		IMDB		BNC	
	LDA	VRTM	LDA	VRTM	LDA	VRTM
5	1.61	0.91	1.60	0.96	1.61	1.26
10	2.29	1.65	2.29	1.56	2.30	1.76
15	2.70	1.69	2.71	2.10	2.71	1.77
30	3.39	2.23	3.39	2.54	3.39	2.22
50	3.90	2.63	3.90	2.74	3.90	2.64

(b) Average document-level topic θ entropy, across three runs, on the test sets. Lower entropy means a document prefers using fewer topics.

Table 2: We provide both SwitchP [24] results and entropy analysis of the model. These results support the idea that if topic models capture semantic dependencies, then they should capture the topics well, explain the topic assignment for each word, and provide an overall level of thematic consistency across the document (lower θ entropy).

SwitchP is bounded between 0 (more switching) and 1 (less switching), where higher is better. Despite this simplicity, SwitchP was shown to correlate significantly better with human judgments of "good" topics than coherence (e.g., a coefficient of determination of 0.91 for SwitchP vs 0.49 for coherence). Against a number of other potential methods, SwitchP was consistent in having high, positive correlation with human judgments. For this reason, we use SwitchP. However, like other evaluations that measure some aspect of meaning (e.g., BLEU attempting to measure meaning-preserving translations) comparisons can only be made in like-settings.

In Table 2a we compare SwitchP for our proposed algorithm against a variational Bayes implementation of LDA [4]. We compare to this classic method, which we call LDA VB, since both it and VRTM use variational inference: this lessens the impact of the learning algorithm, which can be substantial [25, 10], and allows us to focus on the models themselves.[7] We note that it is difficult to compare to many recent methods: our other baselines do not explicitly track each word's assigned topic, and evaluating them with SwitchP would necessitate non-trivial, core changes to those methods.[8] We push VRTM to associate select words with topics, by (stochastically) assigning topics. Because standard LDA VB does not handle stopwords well, we remove them for LDA VB, and to ensure a fair comparison, we mask all stopwords from VRTM. First, notice that LDA VB switches

[7]Given its prominence in the topic modeling community, we initially examined Mallet. It uses Collapsed Gibbs Sampling, rather than variational inference, to learn the parameters—a large difference that confounds comparisons. Still, VRTM's SwitchP was often competitive with, and in some cases surpassed, Mallet's.

[8]For example, evaluating the effect of a recurrent component in a model like LDA+LSTM [22] is difficult, since the LDA and LSTM models are trained separately. While the LSTM model does include inferred document-topic proportions θ, these values are concatenated to the computed LSTM hidden state. The LDA model is frozen, so there is no propagation back to the LDA model: the results would be the same as vanilla LDA.

Dataset	#1	#2	#3	#4	#5	#6	#7	#8	#9
APNEWS	dead	washington	soldiers	fund	police	state	car	republican	city
	killed	american	officers	million	death	voted	road	u.s.	residents
	hunting	california	army	bill	killed	voters	line	president	st.
	deaths	texas	weapons	finance	accusing	democrats	rail	campaign	downtown
	kill	residents	blaze	billion	trial	case	trail	candidates	visitors

Table 3: Nine random topics extracted from a 50 topic VRTM learned on the APNEWS corpus. See Table A2 in the Appendix for topics from IMBD and BNC.

Data	Generated Sentences
APNEWS	• a damaged car and body <unk> were taken to the county medical center from dinner with one driver. • the house now has released the <unk> of $ 100,000 to former <unk> freedom u.s. postal service and $ <unk> to the federal government. • another agency will investigate possible abuse of violations to the police facility . • not even if it represents everyone under control . we are getting working with other items . • the oklahoma supreme court also faces a maximum amount of money for a local counties. • the money had been provided by local workers over how much <unk> was seen in the spring. • he did n't offer any evidence and they say he was taken from a home in front of a vehicle.
IMDB	• the film is very funny and entertaining . while just not cool and all ; the worst one can be expected. • if you must view this movie , then i 'd watch it again again and enjoy it .this movie surprised me . • they definitely are living with characters and can be described as vast <unk> in their parts . • while obviously not used as a good movie , compared to this ; in terms of performances . • i love animated shorts , and once again this is the most moving show series i 've ever seen. • i remember about half the movie as a movie . when it came out on dvd , it did so hard to be particularly silly . • this flick was kind of convincing . john <unk> 's character better could n't be ok because he was famous as his sidekick.
BNC	• she drew into her eyes . she stared at me . molly thought of the young lady , there was lack of same feelings of herself. • these conditions are needed for understanding better performance and ability and entire response . • it was interesting to give decisions order if it does not depend on your society , measured for specific thoughts , sometimes at least . • not a conservative leading male of his life under waste worth many a few months to conform with how it was available . • their economics , the brothers began its $ <unk> wealth by potential shareholders to mixed them by tomorrow . • should they happen in the north by his words , as it is a hero of the heart , then without demand . • we will remember the same kind of the importance of information or involving agents what we found this time.

Table 4: Seven randomly generated sentences from a VRTM model learned on the three corpora.

approximately twice as often. Second, as intuition may suggest, we see that with larger number of topics both methods' SwitchP decreases—though VRTM still switches less often. Third, we note sharp decreases in LDA VB in going from 5 to 10 topics, with a more gradual decline for VRTM. As Lund et al. [24] argue, these results demonstrate that our model and learning approach yield more "consistent" topic models; this suggests that our approach effectively summarizes thematic words consistently in a way that allows the overall document to be modeled.

Inferred Document Topic Entropy To better demonstrate characteristics of our learned model, we report the average document topics entropy with $\alpha = 0.5$. Table 2b presents the results of our approach, along with LDA VB. These results show that the sparsity of relevant/likely topics are much more selective about which topics to use in a document. Together with the SwitchP and perplexity results, this suggests VRTM can provide consistent topic analyses.

Topics & Sentences To demonstrate the effectiveness of our proposed masked embedding representation, we report nine random topics in Table 3 (see Table A2 in the appendix for more examples), and seven randomly generated sentences in Table 4. During the training we observed that the topic-word distribution matrix (β) includes stop words at the beginning and then the probability of stop words given topics gradually decreases. This observation is consistent with our hypothesis that the stop-words do not belong to any specific topic. The reasons for this observation can be attributed to the uniform distributions defined for variational parameters, and to controlling the updates of β matrix while facing the stop and non-stop words. A rigorous, quantitative examination of the generation capacity deserves its own study and is beyond the scope of this work. We provide these so readers may make their own qualitative assessments on the strengths and limitations of our methods.

5 CONCLUSION

We incorporated discrete variables into neural variational without analytically integrating them out or reparametrizing and running stochastic backpropagation on them. Applied to a recurrent, neural topic model, our approach maintains the discrete topic assignments, yielding a simple yet effective way to learn thematic vs. non-thematic (e.g., syntactic) word dynamics. Our approach outperforms previous approaches on language understanding and other topic modeling measures.

Broader Impact

The model used in this paper is fundamentally an associative-based language model. While NVI does provide some degree of regularization, a significant component of the training criteria is still a cross-entropy loss. Further, this paper's model does not examine adjusting this cross-entropy component. As such, the text the model is trained on can influence the types of implicit biases that are transmitted to the learned syntactic component (the RNN/representations h_t), the learned thematic component (the topic matrix β and topic modeling variables θ and z_t), and the tradeoff(s) between these two dynamics (l_t and ρ). For comparability, this work used available datasets that have been previously published on. Based upon this work's goals, there was not an in-depth exploration into any biases within those datasets. Note however that the thematic vs. non-thematic aspect of this work provides a potential avenue for examining this. While we treated l_t as a binary indicator, future work could involve a more nuanced, gradient view.

Direct interpretability of the individual components of the model is mixed. While the topic weights can clearly be inspected and analyzed directly, the same is not as easy for the RNN component. While lacking a direct way to inspect the *overall* decoding model, our approach does provide insight into the thematic component.

We view the model as capturing thematic vs. non-thematic dynamics, though in keeping with previous work, for evaluation we approximated this with non-stopword vs. stopword dynamics. Within topic modeling stop-word handling is generally considered *simply* a preprocessing problem (or obviated by neural networks), we believe that preprocessing is an important element of a downstream user's workflow that is not captured when preprocessing is treated as a stand-alone, perhaps boring step. We argue that future work can examine how different elements of a user's workflow, such as preprocessing, can be handled with our approach.

Acknowledgements and Funding Disclosure We would like to thank members and affiliates of the UMBC CSEE Department, including Edward Raff, Cynthia Matuszek, Erfan Noury and Ahmad Mousavi. We would also like to thank the anonymous reviewers for their comments, questions, and suggestions. Some experiments were conducted on the UMBC HPCF. We'd also like to thank the reviewers for their comments and suggestions. This material is based in part upon work supported by the National Science Foundation under Grant No. IIS-1940931. This material is also based on research that is in part supported by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003. The U.S.Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the Air Force Research Laboratory (AFRL), DARPA, or the U.S. Government.

References

[1] Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A compressed sensing view of unsupervised text embeddings, bag-of-n-grams, and lstms. In *International Conference on Learning Representations*, 2018.

[2] Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. Nonparametric spherical topic modeling with word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016.

[3] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[5] Shammur Absar Chowdhury and Roberto Zamparelli. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

[6] Steven P Crain, Shuang-Hong Yang, Hongyuan Zha, and Yu Jiao. Dialect topic modeling for improved consumer medical search. In *AMIA Annual Symposium Proceedings*, volume 2010, page 132. American Medical Informatics Association, 2010.

[7] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015.

[8] Adji B. Dieng, Chong Wang, Jianfeng Gao, and John W. Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. In *5th International Conference on Learning Representations, ICLR*, 2017.

[9] Jacob Eisenstein, Amr Adel Hassan Ahmed, and Eric P. Xing. Sparse additive generative models of text. In *ICML*, 2011.

[10] Francis Ferraro. *Unsupervised Induction of Frame-Based Linguistic Forms*. PhD thesis, Johns Hopkins University, 2017.

[11] Olivier Ferret. How to thematically segment texts by using lexical cohesion? In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, 1998.

[12] Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pages 441–452, 2018.

[13] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkEYojRqtm.

[14] Pankaj Gupta, Yatin Chaudhary, Florian Buettner, and Hinrich Schütze. Texttovec: Deep contextualized neural autoregressive topic models of language with distributed compositional prior. *arXiv preprint arXiv:1810.03947*, 2018.

[15] Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. Variational pretraining for semi-supervised text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[16] Eva Hajičová and Jiří Mírovský. Discourse coherence through the lens of an annotated text corpus: A case study. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1259.

[17] Geoffrey E Hinton and Ruslan R Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614, 2009.

[18] Weonyoung Joo, Wonsung Lee, Sungrae Park, , and Il-Chul Moon. Dirichlet variational autoencoder, 2019. URL https://openreview.net/forum?id=rkgsvoA9K7.

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[20] Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. The emergence of number and syntax units in LSTM language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1002.

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

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

[23] Shuangyin Li, Yu Zhang, Rong Pan, Mingzhi Mao, and Yang Yang. Recurrent attentional topic model. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[24] Jeffrey Lund, Piper Armstrong, Wilson Fearn, Stephen Cowley, Courtni Byun, Jordan L. Boyd-Graber, and Kevin D. Seppi. Automatic evaluation of local topic quality. In *ACL*, 2019.

[25] Chandler May, Francis Ferraro, Alan McCree, Jonathan Wintrode, Daniel Garcia-Romero, and Benjamin Van Durme. Topic identification and discovery on text and speech. In *EMNLP*, 2015.

[26] Paola Merlo and Suzanne Stevenson. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):373–408, 2001.

[27] Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736, 2016.

[28] Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239. IEEE, 2012.

[29] Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, and Marc'Aurelio Ranzato. Learning longer memory in recurrent neural networks. *arXiv preprint arXiv:1412.7753*, 2014.

[30] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, 2011.

[31] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *Proceedings of the 31st International Conference on International Conference on Machine Learning-Volume 32*, pages II–1791. JMLR. org, 2014.

[32] Seyedahmad Mousavi, Mehdi Rezaee, Ramin Ayanzadeh, et al. A survey on compressive sensing: Classical results and recent advancements. *arXiv: 1908.01014*, 2019.

[33] Christian A Naesseth, Francisco JR Ruiz, Scott W Linderman, and David M Blei. Reparameterization gradients through acceptance-rejection sampling algorithms. *arXiv preprint arXiv:1610.05683*, 2016.

[34] Ramesh Nallapati, Igor Melnyk, Abhishek Kumar, and Bowen Zhou. Sengen: Sentence generating neural variational topic model. *CoRR*, abs/1708.00308, 2017. URL http://arxiv.org/abs/1708.00308.

[35] Anh Tuan Nguyen, Tung Thanh Nguyen, Tien N Nguyen, David Lo, and Chengnian Sun. Duplicate bug report detection with a combination of information retrieval and topic modeling. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pages 70–79. ACM, 2012.

[36] Michael John Paul. *Topic Modeling with Structured Priors for Text-Driven Science*. PhD thesis, Johns Hopkins University, 2015.

[37] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[38] Joseph Reisinger, Austin Waters, Bryan Silverthorn, and Raymond J Mooney. Spherical topic models. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 903–910, 2010.

[39] Mehdi Rezaee, Francis Ferraro, et al. Event representation with sequential, semi-supervised discrete variables. *arXiv: 2010.04361*, 2020.

[40] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[41] Pannawit Samatthiyadikun and Atsuhiro Takasu. Supervised deep polylingual topic modeling for scholarly information recommendations. In *ICPRAM*, pages 196–201, 2018.

[42] Akash Srivastava and Charles A. Sutton. Autoencoding variational inference for topic models. In *ICLR*, 2017.

[43] Nitish Srivastava, Ruslan Salakhutdinov, and Geoffrey Hinton. Modeling documents with a deep boltzmann machine. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, pages 616–624, Arlington, Virginia, United States, 2013. AUAI Press. URL http://dl.acm.org/citation.cfm?id=3023638.3023701.

[44] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[45] Shawn Tan and Khe Chai Sim. Learning utterance-level normalisation using variational autoencoders for robust automatic speech recognition. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 43–49. IEEE, 2016.

[46] Mirwaes Wahabzada, Anne-Katrin Mahlein, Christian Bauckhage, Ulrike Steiner, Erich-Christian Oerke, and Kristian Kersting. Plant phenotyping using probabilistic topic models: uncovering the hyperspectral language of plants. *Scientific reports*, 6:22482, 2016.

[47] Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981, 2009.

[48] Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. Topic compositional neural language model. In *AISTATS*, 2017.

[49] 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: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 166–177, 2019.

[50] Xiaogang Wang. Action recognition using topic models. In *Visual Analysis of Humans*, pages 311–332. Springer, 2011.

[51] Tsung-Hsien Wen and Minh-Thang Luong. Latent topic conversational models. *arXiv preprint arXiv:1809.07070*, 2018.

[52] Yinfei Yang, Forrest Bao, and Ani Nenkova. Detecting (un)important content for single-document news summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017.

[53] Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. Whai: Weibull hybrid autoencoding inference for deep topic modeling. In *ICLR*, 2018.

Supplementary Material

A Proof of Theorm 1

Theorem 1 states: If in the generative story $\rho = 0$, and we have a uniform distribution for the prior and variational topic posterior approximation $\big(p(\theta) = q(\theta|\mathbf{w}, \mathbf{l}) = 1/K\big)$, then VRTM reduces to a Recurrent Neural Network to just reconstruct the words given previous words.

Following is the proof.

Proof. If $\rho = 0$ then for every token t, $l_t = 0$. Therefore, the reconstruction term simplifies to $\mathcal{L}_w = \sum_{t=1}^{T} \log p(w_t; h_t)$. We proceed to analyze the other remaining terms as following. The first terms on the right-hand side of $\mathcal{L}_z$ and $\mathcal{L}_\phi$ are over thematic word, and simply we have $\mathcal{L}_z + \mathcal{L}_\phi = 0$. Since all the tokens are forced to have the same label, the classifier term is $\mathcal{L}_l = \sum_{t=1}^{T} \log p(l_t; h_t) = 0$. Also $\mathcal{L}_\theta = 0$, since it is the KL divergence between two equal distributions. Overall, $\mathcal{L}$ reduces to $\sum_{t=1}^{T} \log p(w_t; h_t)$, indicating that the model is just maximizing the log-model evidence based on the RNN output. □

B Dataset Details

Table A1 provides details on the different datasets we use. Note that these are the same datasets as have been used by others [49].

C Implementation Details

For an RNN we used a single-layer LSTM with 600 units in the hidden layer, set the size of embedding to be 400, and had a fixed/maximum sequence length of 45. We also present experiments demonstrating the performance of basic RNN and GRU cells in Table 1a. Note that although our embedding size is higher, we are use an end-to-end training manner without using pre-trained word embeddings. However, as shown in Table 1a, we also examined the impact of using lower dimension embeddings and found our results to be fairly consistent. We found that using pretrained word embeddings such as word2vec could result in slight degradations in perplexity, and so we opted to learn the embeddings from scratch. We used a single Monte Carlo sample of θ per document and epoch. We used a dropout rate of 0.4. In all experiments, α is fixed to 0.5 based on the validation set metrics. We optimized our parameters using the Adam optimizer with initial learning rate 10^{-3} and early stopping (lack of validation performance improvement for three iterations). We implemented the VRTM with Tensorflow v1.13 and CUDA v8.0. Models were trained using a single Titan GPU. With a batch size of 200 documents, full training and evaluation runs typically took between 3 and 5 hours (depending on the number of topics).

D Perplexity Calculation

Following previous efforts we calculate perplexity across D documents with N tokens total as

$$\text{perplexity} = \exp\left(\frac{-\sum_{d=1}^{D} \log p(\mathbf{w}_d)}{N}\right), \tag{6}$$

Dataset	Vocab	Training			Development			Testing		
		# Docs	# Sents	# Tokens	# Docs	# Sents	# Tokens	# Docs	# Sents	# Tokens
APNEWS	$7,788$	$50K$	$0.7M$	$15M$	$2K$	$27.4K$	$0.6M$	$2K$	$26.3K$	$0.6M$
IMDB	$8,734$	$75K$	$0.9M$	$20M$	$12.5K$	$0.2M$	$0.3M$	$12.5K$	$0.2M$	$0.3M$
BNC	$9,769$	$15K$	$0.8M$	$18M$	$1K$	$44K$	$1M$	$1K$	$52K$	$1M$

Table A1: A summary of the datasets used in our experiments. We use the same datasets and splits as in previous work [49].

where $\log p(\mathbf{w}_d)$ factorizes according to eq. 1. We approximate the marginal probability of each token $p(w_t; \beta, h_t)$ within d as

$$
\begin{aligned}
p(w_t; \beta, h_t) &= \int p(\theta) \sum_{k=1}^{K} \sum_{l_t=0}^{1} p(w_t | z_t = k, l_t; h_t, \beta) p(z_t | l_t, \theta) p(l_t; h_t) \, d\theta. \\
&\approx \frac{1}{S} \sum_{s=1}^{S} \sum_{k=1}^{K} \sum_{l_t=0}^{1} p(w_t | z_t = k, l_t; h_t, \beta) p(z_t = k | l_t, \theta^{(s)}) p(l_t; h_t) \\
&= \frac{1}{S} \sum_{s=1}^{S} \sum_{k=1}^{K} \theta_k^{(s)} p(w_t | z_t = k; h_t, \beta) p(l_t = 1; h_t) + p(w_t; h_t) p(l_t = 0; h_t), \qquad (7)
\end{aligned}
$$

Each $\theta^{(s)} \sim q(\theta | w_{1:T}, \gamma)$ is sampled from the computed posterior approximation, and we draw S samples per document.

E Text Generation

The overall generating document procedure is illustrated in Algorithm 1. We use <SEP> as a special symbol that we silently prepend to sentences during training.

Algorithm 1 Generating Text

Input: sequence length (l)
Output: generated sentence
  $i \leftarrow 0$
  $w_0 \leftarrow$ <SEP>
  $\mathbf{w} \leftarrow [w_0]$
  repeat
    $i \leftarrow i + 1$
    $\theta^{(s)} \sim q(\theta | \mathbf{w})$
    $w_i \sim p(w_i | \mathbf{w})$
    $\mathbf{w} \leftarrow [\mathbf{w}, w_i]$
  until $i < l$
  return $\mathbf{w}$

We limit the concatenation step ($\mathbf{w} \leftarrow [\mathbf{w}, w_i]$) to the previous 30 words.

F Generated Topics

See Table A2 for additional, randomly sampling topics from VRTM models learned on APNEWS, IMDB, and BNC (50 topics).

G Generated Sentences

In this part we provide some sample, generated output explain how we can generate text using VRTM. To this end, we begin with the start word and then we proceed to predict the next word given all the previous words. It is worth mentioning that for this task, the labels of stop and non-stop words are marginalized out and the model is predicting these labels best on RNN hidden states. This conditional probability is

$$
\begin{aligned}
p(w_t | w_{1:t-1}) = \int p(\theta) \sum_{k=1}^{K} \sum_{l_t=0}^{1} & p(w_t | z_t, l_t; h_t, \beta) \\
& p(z_t | l_t, \theta) p(l_t; h_t) \, d\theta. \qquad (8)
\end{aligned}
$$

Computing Eq. 8 exactly is intractable in our context. We apply Monte Carlo sampling $\theta^{(s)} \sim q(\theta | w_{1:T}, \alpha)$. It is too expensive to recompute θ with each word generated. To alleviate this problem,

Dataset	#1	#2	#3	#4	#5	#6	#7	#8	#9
APNEWS	dead	washington	soldiers	fund	police	state	car	republican	city
	killed	american	officers	million	death	voted	road	u.s.	residents
	hunting	california	army	bill	killed	voters	line	president	st.
	deaths	texas	weapons	finance	accusing	democrats	rail	campaign	downtown
	kill	residents	blaze	billion	trial	case	trail	candidates	visitors
IMDB	films	horror	pretty	friends	script	comedy	funny	hate	writing
	directed	murder	beautiful	series	line	starring	jim	cold	wrote
	story	strange	masterpiece	dvd	point	fun	amazed	sad	fan
	imdb	killing	intense	channel	describe	talking	naked	monster	question
	spoilers	crazy	feeling	shown	attention	talk	laughing	dawn	terribly
BNC	king	house	research	today	letter	system	married	financial	children
	london	st	published	ago	page	data	live	price	played
	northern	street	report	life	books	bit	love	poor	class
	conservative	town	reported	years	bible	runs	gentleman	thousands	12
	prince	club	title	earlier	writing	supply	dance	commission	age

Table A2: Nine random topics extracted from a 50 topic VRTM learned on the APNEWS, IMDB and BNC corpora.

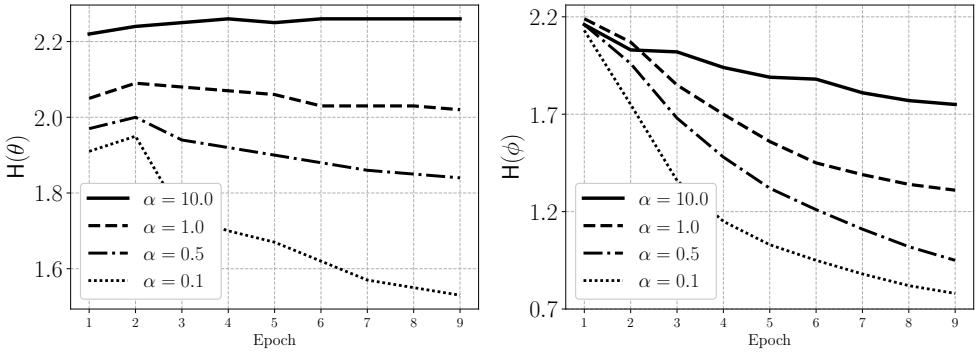


Figure A1: We examine the impact α has on the induced variational approximations q from a 10 topic VRTM (selected due to dev perplexity performance). **Left:** effect of α on $\mathsf{H}(\theta)$, **Right:** Appropriate choice of α also reduces $\mathsf{H}(\phi)$.

we can use a "sliding window" of previous words rather than the whole sequence to periodically resample θ [28, 8]. In this, we maintain a buffer of generated words and resample θ when the buffer is full (at which point we empty it and continue generating). We found that splitting each training document into blocks of 10 sentences generated more coherent sentence. Table 4 illustrates the quality of some generated sentences.

G.1 The Effect of Hyperparameters on Topic Selectivity

Following previous work in language modeling that take the sparsity into consideration [1, 32], we sought for a selective token topic assignment with low entropy. First, we slightly overload the notation of entropy to define an average of non stop word-topics entropy and similarly document entropy: $\mathsf{H}(\phi) = -\frac{1}{T_n}\sum_{t:l_t=1}\sum_{k=1}^{K}\phi_t^k \log \phi_t^k$, and $\mathsf{H}(\theta) = -\frac{1}{T_d}\sum_{k=1}^{K}\theta^k \log \theta^k$, where T_n and T_d are the total number of non-stop words in the document and the total number of documents, respectively. Second, the Dirichlet distribution can easily be parameterized to generate sparse samples. Now we provide some intuitive explanation. In our setting, the equality $\mathcal{L}_z + \mathcal{L}_\phi = \frac{1}{S}\sum_{s=1}^{S}\sum_{t:l_t=1}\sum_{k=1}^{K}\phi_t^k \log \frac{\theta_k^{(s)}}{\phi_t^k}$ holds, which is the (negative) KL-divergence between ϕ and θ parameters. Moreover, on the $\mathcal{L}_\theta$ side, $\theta^{(s)}$ samples are controlled by the prior distribution. Overall, selectivity of ϕ strongly depends on the choice of prior parameters. As shown in Fig. A1, not only we can control the value of $\mathsf{H}(\theta)$, but also we can reduce $\mathsf{H}(\phi)$ by tuning the prior parameters without the need of any other regularizer.