Topic Modeling Revisited: A Document Graph-based Neural Network Perspective

Anonymous Author(s)

Abstract

Traditional topic modeling usually has the bag-of-words assumption, where each word is required to be conditionally independent in the same document. As a result, the semantic dependency among words is ignored in the process of the generative story and the topic formulation. However, the semantic dependency among words, if properly measured, can be exploited for improving the semantic comprehension and model interpretability. To this end, in this paper, we transform each document into a directed graph with word dependency as edges between word nodes and develop a Graph Neural Topic Modeling (GNTM) method. Specifically, in GNTM, a well-defined probabilistic generative story is designed to model both the graph structure and word sets with multinomial distributions on the vocabulary and word dependency edge set as the topics. Meanwhile, an Autoencoder Variational Inference (AVI) approach is proposed to learn our model with graph neural networks to encode the document graphs. Moreover, we theoretically show that Latent Dirichlet Allocation (LDA) can be derived from GNTM as a special case with similar objective functions. Finally, extensive experiments on four benchmark datasets have clearly demonstrated the effectiveness and interpretability of GNTM compared with state-of-the-art baselines.

1 Introduction

As one of the most widely-used techniques for document analysis, topic modeling aims to learn a set of latent topics from the observed document collection, each of which describes an interpretable semantic concept. In particular, Latent Dirichlet Allocation (LDA) [5] and its extensions [3, 9, 40, 51] have achieved great success in various application scenarios over the past decades. These approaches usually specify a probabilistic generative model to draw the document data with a structure of latent variables sampled from prior distributions and word distribution on the vocabulary as topics. Recently, with the development of Autoencoder Variational Inference (AVI) [22], a stochastic variational inference algorithm with neural networks for topic modeling, namely Neural Topic Model, has attracted great attention [31, 30, 11, 12, 38, 42, 52] , due to its appealing flexibility and scalability.

However, it is well-known that traditional topic modeling usually has the bag-of-word assumption, where the words in a document are assumed to be "exchangeable", which also arouses two typical challenges behind most topic models. On one hand, given the topic assignment of each word in the document, the words are conditionally independent. This implicates that the dependency relationship among words is totally ignored, while words are dependent on each other according to linguistic knowledge [27]. On the other hand, the concept of topics is introduced as multinomial distributions on vocabulary without modeling the dependency relation among words, which produces more ambiguous terms in topics. For example, LDA discovers word such as "station" in a topic which does not seem to be that insightful. Instead, we can convey more interpretability to readers by mining the strong word dependency between "space" and "station" [17].

Along this line, many extensions have been proposed to address the above two issues by dealing with the dependency relation among words. One thread of these works focus on the sequential dependency by taking the document as a sequence of words [12, 13, 14, 25, 28, 38, 48, 44, 47, 51]. However, most of them prefer to enhance language modeling with better performance on language modeling

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

metric, such as Perplexity, rather than the state-of-the-art quality of learned topics. It may be due to the limitation that they only handle the linear dependency of words [27] and push models to capture the local syntax information instead of global semantic information. Specifically, to mitigate the first problem, most of them assume that the topic assignment or generative possibility of a word is dependent on the proceeding sentence [12, 13, 14, 25, 38, 48]. As for the second problem, related literature is restricted to exploring n-gram terms by integrating few neighboring words as one term with Markov Chain based generative story [44, 47, 28].

Nonetheless, words may be mutually dependent on each other in a much more complex and non-linear manner. Therefore, several studies explore to model the graph representation of documents, where nodes denote words and edges represent the relationships among words, such as syntax or semantic relation constructed by dependency parsing [27, 46] or justified by relative position among words [9, 52]. They either consider the graph structure as side information to constrain the relation of topic assignments of words [27, 46], or replacing the word sets with edge sets as the instances to generative [9, 52]. Most of them only trickle with the first problem without efforts to enhance the interpretability of topics.

To this end, in this paper, we represent documents as directed semantic graphs by introducing word dependency as edges between word nodes, and develop a Graphic Neural Topic Modeling (GNTM) method. The key contributions of our model can be categorized into four parts: 1) we formulate a well-defined probabilistic generative story of the document graph with the novel generative probabilistic functions for both graph structure and word sets; 2) we propose a new concept of topics, consisting of multinomial distributions on both the vocabulary, like conventional topic models, and word dependency edges; 3) an AVI inference approach is designed to infer our model with graph neural network to encode the document graphs; 4) we also demonstrate that LDA can be derived from GNTM as a special case with similar objective functions. Finally, extensive experiments over four benchmark datasets have clearly demonstrated the effectiveness and interpretability of GNTM compared with state-of-the-art baselines.

2 Preliminary

2.1 Topic Model

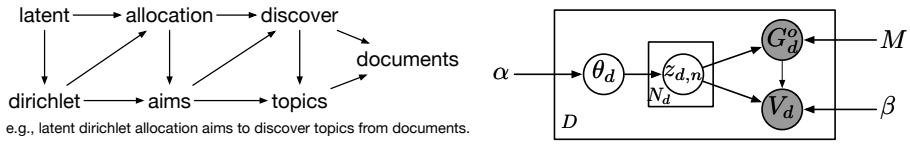
Topic modeling is one of the most important text mining techniques and has been extensively studied for a variety of applications in recent two decades [2, 19]. Among the most representative models, Latent Dirichlet Allocation (LDA) [5] formulates each topic k as the distribution β_k^v on vocabulary V and draws each word $w_{d,n}$ in a document d from one of K topics stochastically. The well-known generative process of LDA can be described as three steps: 1) sampling document-topic proportions $\theta_d \sim \mathrm{Dir}(\alpha)$, where α is hype-parameter for the Dirichlet distribution prior; 2) drawing individual topic assignments $z_{d,n} \sim \mathrm{Multi}(\theta_d)$; 3) generating each word $w_{d,n} \sim \mathrm{Multi}(\beta^v_{z_{d,n}})$. Then, the marginal likelihood for a document d is:

$$p(d|\alpha, \beta) = \int_{\theta_d} p(\theta_d|\alpha) \prod_n \sum_{z_{d,n}} p(w_{d,n}|\beta^v_{z_{d,n}}) p(z_{d,n}|\theta_d) d\theta_d. \tag{1}$$

Conventional approaches for inferring topic models, such as LDA, are either Markov Chain Monte Carlo [7] or probabilistic variational inference [20]. However, they either require careful selection of distribution for each latent variable or tolerate arduous and customized mathematical derivation case by case, which limits the flexibility and scalability of model design [31]. Recently, with the successes of Autoencoder Variational Inference (AVI) [22], which applies the variational distributions parameterized by neural networks to approximate the posterior of latent variables, the use of deep learning on inferring topic models has attracted more and more attention [49]. Most of those neural topic models (NTMs) [31, 30, 42, 11] can be regarded as extensions of LDA, but often integrate out the discrete latent variable $z_{d,n}$. Therefore, the variational lower bound (ELBO) for the log-likelihood of the document d can be derived as:

$$\mathcal{L}_d = E_{q_\phi(\theta|d)}[\sum_n \log p(w_{d,n}|\theta_d, \beta^v)] - D_{KL}[q_\phi(\theta_d|d)||p(\theta_d)], \tag{2}$$

where $q_\phi(\theta|d)$ is the variational distribution of θ_d parametrized by a neural network ϕ. To obtain a differentiable estimator of $\mathcal{L}_d$, the reparameterization trick (RT) has been applied for θ. In particular, due to the limitation of RT on Dirichlet distribution [42], those approaches usually stand on Gaussian-based posterior $q_\phi(\theta_d|d)$.



(a) Document graph example with $s = 2$ (b) Directed probabilistic graph

Figure 1: The document graph example and the directed probabilistic graph of our model.

2.2 Document Graph Construction

Most existing methods in topic models commonly represent a document as an unordered collection of words, known as bag-of-word (BOW) representation, which totally ignores the relationship among words in the document. To represent documents with word dependency relations, we discuss the graph representation of documents here.

Formally, given one document d with word sequences $[w_{d,t}]_{t=1}^{N_d}$, we construct a directed edge from w_{d,t_1} to w_{d,t_2} if $0 < t_2 - t_1 \leq s$, where s is the hyper-parameter of window size. The hidden logic is that the semantic is consistent and interdependent among neighboring words, which can be traced back to n-gram [8], BTM [9] and Word2vec [32]. Then, we can donate a document d as a graph $G_d = (V_d, E_d)$, where $V_d = \{w_{d,n}\}_{n=1}^{N_d}$ (words as nodes) and E_d are sets of nodes and word edges, respectively. Note that, for simplification in analysis, we distinguish the same words on different positions in E_d and V_d. We also denote E as the unique word edge sets in the whole document collection with nodes on the vocabulary $V = \{w_i\}_{i=1}^{|V|}$ (unique word set). For convention, we further denote $G_d^o = (V_d^o, E_d^o)$ as the graph structure of G_d with only the placeholders $V_d^o = \{1, 2, ..., N_d\}$, not the specific words V_d, where $|V_d^o| = |V_d|$, and the placeholder edge set E_d^o is the edge set on the placeholder set V_d^o, distinguishing from the word edge set E_d. The Figure 1(a) shows one document graph on a sentence. We also filter out the stop words, and words and dependency edges with low frequency among the whole collection to reduce noise.

3 Graph Neural Topic Model

With the definition of the document graph G_d, we turn to introduce the technical detail of GNTM. Specifically, we first formulate the generative story for both the graph structure G_d^o and word sets V_d. Then, an AVI inference algorithm will be proposed to learn our model.

3.1 Generative Story

Figure 1(b) shows our directed graphical model. In contrast to three-step generative process in LDA, four steps are involved in the generative story. The main differences lie in the conditional generation for the observed data given all latent variables (steps 3 and 4 in the following). To be specific, 1) the generative story of a document d begins with topic proportion θ drawn from the Dirichlet prior $\mathrm{Dir}(\alpha)$. Then, 2) each word $w_{d,n}$ (or placeholder n) has corresponding topic assignment $z_{d,n}$ drawn from $\mathrm{Multi}(\theta)$. Finally we generate the observed data with two steps: 3) drawing the graph structure G_d^o conditioning on the topic assignment set Z_d, i.e., $G_d^o \sim p(G_d^o|Z_d)$; 4) generating the word set V_d based on both topic assignment set Z_d and graph structure G_d^o, i.e., $V_d \sim p(V_d|G_d^o, Z_d)$. Based on the generative process, the joint model of G_d and all latent variables can be decomposed as:

$$p(G_d, \theta_d, Z_d; \alpha) = p(V_d|Z_d, G_d^o)p(G_d^o|Z_d) \prod_{n=1}^{N_d} p(z_{d,n}|\theta_d)p(\theta|\alpha). \tag{3}$$

In the following, we turn to formulate the formal definition of $p(G_d^o|Z_d)$ and $p(V_d|Z_d, G_d^o)$ with additional parameters, i.e., topic dependency matrix M and topic set β, which is the most important contribution of our model.

3.1.1 Generation for Graph Structure

Here, we aim to construct the graph structure G_d^o just based on the topic assignment $z_{d,n}$ on each placeholder $w_{d,n}^o$. The hidden motivation is to explain the word dependency as the semantic dependency among their topics, which can encourage our model to find the relation among different topics in turn. To be specific, we denote $M \in \mathbb{R}^{K \times K}$ as the dependency matrix among topics, where each

element $m_{i,j} \in [0,1]$ represents the possibility that there exists a dependency edge between two placeholders with topic i and j, respectively. Then, we can formulate $p(G_d^o|Z_d)$ as :

$$p(G_d^o|Z_d;M) = \prod_{(n,n')\in E_d^o} m_{z_{d,n},z_{d,n'}} \prod_{(n,n')\notin E_d^o} (1 - m_{z_{d,n},z_{d,n'}}), \tag{4}$$

where M is a learnable parameter without additional assumptions.

3.1.2 Generation for Word Set

Contrast to LDA, once a document is represented as a graph, the word sampling should be cast into the sampling of this graph. In other words, we should consider dependency between words when generating word set V_d. To provide a well-defined word sampling process, we first re-define the concept of topic. Different from the conventional concept of topic in LDA, which is represented by the multinomial distribution β_k^v on the vocabulary, we further enhance each topic k by introducing one distribution on word dependency edge with parameters β_k^e, i.e., $\beta_k = (\beta_k^v, \beta_k^e)$, where:

$$\sum_{w\in V} \beta_{k,w}^v = 1, \quad \sum_{w,w'\in V} \beta_{k,w}^v \beta_{k,w'}^v \beta_{k,(w,w')}^e = 1, \ \beta_{k,w}^v \geq 0, \ \beta_{k,(w,w')}^e \geq 0. \tag{5}$$

where the product $\beta_{k,w}^v \beta_{k,w'}^v \beta_{k,(w,w')}^e$ denotes the possibility of the directed edge from word w to w' under the topic k. Note that, in practice, only parameters $\beta_{k,(w,w')}^e$ corresponding to edge $(w.w') \in E$ would be used in model learning. In other words, we can reduce the space complexity of β_k^e into $O(|E|)$ easily by assuming that $\beta_{k,(w,w')}^e = 0, \forall (w,w') \notin E$.

Then, based on the new concept of the topics, we propose a well-defined probability measure for generating each word set V_d with the document graph structure G_d^o and topic assignment sets Z_d. The probability of $p(V_d|Z_d, G_d^o)$ is modeled by :

$$\begin{aligned} p(V_d|Z_d, G_d^o;\beta) &= \prod_{k=1}^{K} p(V_{d,k}|G_{d,k}^o, \beta_k), \\ p(V_{d,k}|G_{d,k}^o;\beta_k) &= \frac{1}{|E_{d,k}|} \prod_{w\in V_{d,k}} \beta_{k,w}^v \sum_{(w,w')\in E_{d,k}} \beta_{k,(w,w')}^e, \end{aligned} \tag{6}$$

where $G_{d,k}^o$ is the graph structure consisting of all placeholders with the topic assignment k, similarly, $V_{d,k}$ is the set of words assigned by topic k and $E_{d,k}$ is the word dependency edge among them. In other words, we split the document graph G_d into K dependent sub-graphs $G_{d,k}$ based on topic assignment set Z_d and then define the generative possibility of each topical sub-graph $G_{d,k}$ with the word dependency edges $E_{d,k}$. If $E_{d,k} = \emptyset$, we set $\sum_{(w,w')\in E_{d,k}} \beta_{k,(w,w')}^e / |E_{d,k}| = 1$. Mathematically, we have the following theorem:

Theorem 1 *Given the topic set β in Equation 5 and the document graph structure G_d^o, the probability function $p(V_d|Z_d, G_d^o;\beta)$ in Equation 6 is a legal probability measure on the vocabulary V.*

The proof can be found in Appendix which is mainly inspired by [27].

3.2 AVI Inference

Here, we infer our model by applying AVI inference algorithm, which is well-know of flexibility and scalability [22]. Based on the mean field assumption, we approximate the posterior of latent variables θ_d and Z_d for each document d with the following variational family:

$$q(\theta_d, Z_d|G_d) = q_\phi(\theta_d|G_d;\mu_d,\delta_d) \prod_{n=1}^{N_d} q_\phi(z_{d,n}|G_d, w_{d,n};\varphi_{d,n}), \tag{7}$$

where $q_\phi(z_{d,n}|G_d, w_{d,n})$ is defined as one multinomial distribution with parameters $\varphi_{d,n}$ and $q_\phi(\theta_d|G_d)$ is the approximation for the true Dirichlet posterior $p(\theta|G_d)$. Both of them are parameterized by neural networks ϕ. However, directly approaching the Dirichlet distributions would make the reparameterization trick hard to apply. Here, we follow [42, 15] and resolve this issue by

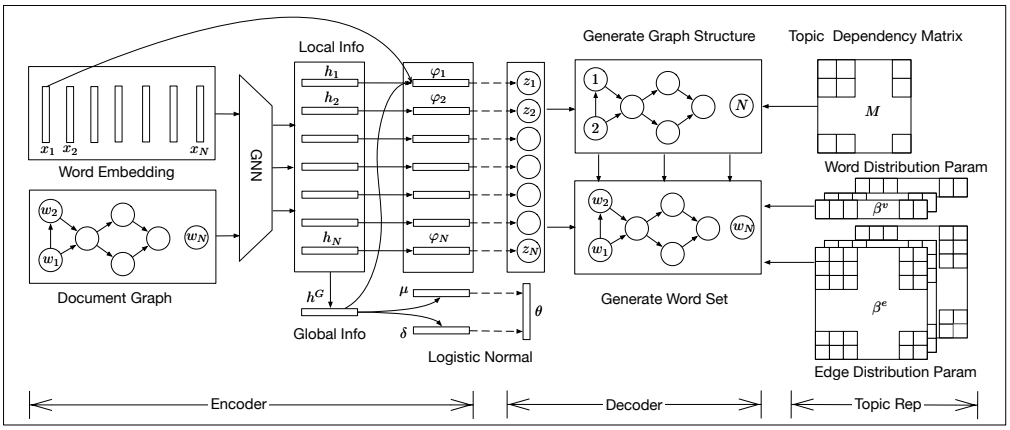


Figure 2: The overivew of network structures to infer our model.

constructing a Laplace approximation to the Dirichlet posterior with the Logistic Normal distribution parameterized by (μ_d, δ_d), that is,

$$\theta_d = \text{softmax}(\mathcal{N}(\mu_d, \delta_d^2)). \tag{8}$$

Then, we can derive the training objective for each document d, i.e., ELBO:

$$\begin{aligned} \mathcal{L}_d = E_{q(Z_d|G_d)}[\log p(G_d^o|Z_d; M) + \log p(V_d|Z_d, G_d^o; \beta)] \\ -KL[q(\theta_d|G_d)|p(\theta_d)] - E_{q(\theta_d|G_d)}[\sum_{n=1}^{N_d} KL[q(z_{d,n}|G_d, w_{d,n})||p(z_{d,n}|\theta_d)]], \end{aligned} \tag{9}$$

where the first line aims to reconstruct the observed data G_d, and the second line consists of two KL divergence that attempts to close the distance between the posterior and prior of latent variables θ_d and Z_d. Fortunately, $KL[q(\theta_d|G_d)|p(\theta_d)]$ and $KL[q(z_{d,n}|G_d, w_{d,n})||p(z_{d,n}|\theta_d)]$ are both analytic (see Appendix). Then, we approximate the expectation of the second KL term over θ with Monte Carlo samples by applying the reparameterization trick on $q(\theta_d|G_d^0)$[1]. As for the deduction of the first line, we will discuss in Section 3.2.3.

After the probabilistic definition of the generative story and the corresponding variational family, we turn to introduce the detailed network structure to implement our model. As the overview shown in Figure 2, the neural network structure of GNTM can be broadly split into three parts: 1) the topic representation, including the topic set β and the dependency relation matrix M; 2) an encoder, which maps the observed data to the variational parameters; 3) a decoder, which computes the possibility of the observed data given the latent variables. We describe each term as below.

3.2.1 Topic Representation

Inspired by [30], we introduce word vectors $r \in \mathbb{R}^{|V| \times H}$, topic-word vector $u^v \in \mathbb{R}^{K \times H}$, topic-edge vector $u^e \in \mathbb{R}^{K \times 2H}$, and generate β_k^v and β_k^e by:

$$\beta_{k,w}^v = \frac{exp(r_w \cdot (u_k^v)^T)}{\sum_{w' \in V} exp(r_{w'} \cdot (u_k^v)^T)}, \ \beta_{k,(w_1,w_2)}^e = \frac{exp(\hat{r}_{w_1,w_2} \cdot (u_k^e)^T)}{\sum_{(w,w') \in E} \beta_{k,w}^v \beta_{k,w'}^v exp(\hat{r}_{w,w'} \cdot (u_k^e)^T)}, \tag{10}$$

where, $\hat{r}_{w_1,w_2} = r_{w_1} \oplus r_{w_2}$. $(\cdot)$ and $\oplus$ denote dot product and concatenation operations, respectively.

Therefore, each topic k can be represented by the vector $u_k = u_k^v \oplus u_k^e$. Meanwhile, β_k^v and β_k^e can be explained as the semantic similarity between the topic and words or edges (the concatenation of word vectors as the edge vector). Along this line, we further define each element of the topic dependency matrix M as the semantic similarity among topics with additional learnable parameter matrix $W \in \mathbb{R}^{3H \times 2Y}$:

$$a_k \oplus b_k = u_k \cdot W, \quad m_{i,j} = \sigma(g(a_i \cdot b_j^T)), \ \ \forall\, i, j, k, \ \ i \neq j, \tag{11}$$

[1]Although, the expectation of the second KL term can also be computed analytically under Dirichlet distribution, in which our model is conjugate, we still use the sampling strategy here, which is more general and can also be applied for more flexible situations without conjugate distributions, such as [4, 45]

where $\sigma(\cdot)$ is the sigmod function. $a_k, b_k \in \mathbb{R}^Y$ are two transitional vectors for the topic k to compute the possibility of directed edges among different topics. $g(\cdot)$ denotes a vector similarity metric and is set as the consine similarity $g(a,b) = C(a \cdot b/(||a||||b||))$ here, in which C is a hyper-parameter to re-scale the range of $\sigma(g(\cdot))$ to approach $[0,1]$.

3.2.2 Encoder

Here, we aim to model the mapping from the observed data G_d to the variational parameters for latent variables θ_d and Z_d. To be specific, we denote $X \in \mathbb{R}^{|V| \times L}$ as the trained and fixed word embeddings for all words in the vocabulary V. For the document d, $X_d \in \mathbb{R}^{|V_d| \times L}$ is the word embedding set for word set V_d. Then, we encode the local information of each node and the global information of the whole graph G_d with one layer of graph neural network, such as GraphConv [34]:

$$\begin{aligned} h_d &= \tanh(GNN(G_d^o, X_d)), \\ h_d^G &= \tanh(\sum_{n=1}^{N_d} \sigma(f_1(h_{d,n} \oplus x_{d,n})) \odot \tanh(f_2(h_{d,n} \oplus x_{d,n}))), \end{aligned} \tag{12}$$

where $h_d \in \mathbb{R}^{|V_d| \times L}$ is the node-level representation vectors, $h_d^G \in \mathbb{R}^L$ is the graph-level representation vector, and $f_*(\cdot)$ is a full connected layer. Self-loops have been added to G_d^o for preserving the current word feature. The second project network is following [26].

Then, the parameters of variational marginals $q(\theta_d|G_d)$ and $q(z_{d,n}|G_d, w_{d,n})$ can be specified as:

$$\begin{aligned} \mu_d = f_\mu(h_d^G), \quad \log \delta_d^2 = f_\delta(h_d^G), \\ \varphi_{d,n} = f_\varphi(h_d^G \oplus h_{d,n} \oplus x_{d,n}). \end{aligned} \tag{13}$$

3.2.3 Decoder

We denote the reconstruction part in Equation 9 (the first line) as $\mathcal{L}_d^R$, which can be deduced as:

$$\begin{aligned} \mathcal{L}_d^R = &\sum_{(n,n') \in E_d^o} \varphi_{d,n}^T \cdot (\log M) \cdot \varphi_{d,n'} + \sum_{(n,n') \notin E_d^o} \varphi_{d,n}^T \cdot \log(1-M) \cdot \varphi_{d,n'} \\ &+ \sum_{n=1}^{N_d} \varphi_{d,n}^T \cdot \beta_{\cdot,w_n}^v + \sum_{k=1}^{K} E_{q(Z_d|G_d)}[\log \frac{\sum_{(n,n') \in E_d^o} z_{d,n}^k z_{d,n'}^k \beta_{k,(w_{d,n}, w_{d,n'})}^e}{\sum_{(n,n') \in E_d^o} z_{d,n}^k z_{d,n'}^k}], \end{aligned} \tag{14}$$

where $z_{d,n}^k = 1$ if $z_{d,n} = k$ otherwise 0. Indeed, the first two terms equal to $E_{q(Z_d|G_d)}[\log p(G_d^o|Z_d)]$ and the other items are derived from $E_{q(Z_d|G_d)}[\log p(V_d|Z_d, G_d^o)]$ (see Appendix) . In practice, the graph G_d^o is often very sparse which causes the unbalance between the first two terms. Here, we follow [23] and re-weight the second terms with $|E_d^o|(\sum_{(n,n') \notin E_d^o} 1)^{-1}$.

Now, the only challenge to compute the ELBO loss $\mathcal{L}_d$ is at the last term in Eqaution 14, which is the expectation of a log-sum function and can not be computed analytically. Here, we estimate this expectation with Monte Carlo samples from $q(Z_d|G_d)$. Meanwhile, to ensure the differentiability, we use Straight-Through Gumbel-Softmax (STGS) estimator [18] to approximate the undifferentiable multinomial sampling, which outputs the discrete samples as the approximated multinomial samples in the neural forward pass, but applies a continuous approximation in the backward pass.

We conclude this section with an observation to show that the objective function in LDA can be derived from that of GNTM as a special case. This demonstrates the generalization of our proposed model and the contribution to model the word dependency by representing documents as graphs. The proof can be found in Appendix.

Corollary 1 *If the word dependency edge E_d of the document d is one empty set, the objective function defined in Equation 9 would reduce into the ELBO in LDA with the Laplace approximation of the Dirichlet distribution.*

4 Experiments

Here, we conduct extensive experiments on four benchmark text datasets to evaluate the effectiveness and interpretability of our model. (Our code will be publicly available after possible publication.)

4.1 Experimental Setup

Datasets: Our experiments were conducted on four benchmark datasets with varying sizes, including 20 News Groups (**20NG**) [24], Tag My News (**TMN**) [43], the British National Corpus (**BNC**) [10], **Reuters** extracted from the Reuters-21578 dataset. The statistics and links of these datasets are shown in Appendix. In particular, 20NG and TMN are single-labeled datasets. We followed [14] and only used the title text in TMN to represent the short documents

Evaluation Metrics: Here, we focus on measuring the quality of learned topics. Traditionally, perplexity has been used to measure the quality of topic models, but it has been repeatedly shown that perplexity is not a good metric for the qualitative evaluation of topics [35]. Therefore, we adopted another two metrics: topic coherence (**TC**) and topic diversity (**TD**). Given the reference corpus, TC measures the interpretability of a topic by computing the semantic coherence in the most significant words [33]. With the suggestion in [39], we selected two TC measurements to provide a robust evaluation, i.e, Normalized Pointwise Mutual Information (NPMI) [1] and C_v [39]. Specifically, both NPMI and C_v are computed on the top words of each topic with the original document corpus of each dataset as the reference documents. Higher value indicates better interpretability. TD aims to measure how diverse the discovered topics are. We followed [11] and defined TD as the percentage of unique words in the top words. TD closes to 0 indicates redundant topics. TD closes to 1 indicates more varied topics. In addition, we also evaluated our model in a document clustering task on labeled datasets, i.e., 20NG and TMN, to evaluate the quality of the learned topic propitiation. Specifically, following [36], we assigned every document the topic with the highest probability as the clustering label and compute **Purity** and Normalized Mutual Information (**NMI**) [41] as metrics. Both of them always range from 0.0 to 1.0, and higher scores reflect better clustering performance.

Benchmark Models: We selected several state-of-the art-topic model as the baselines, including: 1) **LDA** [5]; 2) **GSM** [30], an NTM model which replaces the Dirichlet-Multinomial parametrization in LDA with Gaussian Softmax; 3) **ProdLDA** [42], an NTM model which keeps the Dirichlet-Multinomial parametrization with a Laplace approximation; 4) **ETM** [11], an NTM mdoel which incorporates word embedding to model topics; 5) **GraphBTM** [52], an NTM model with graph-based word dependency; 6) **iDocNADE** [14], an neural autoregressive topic model with sentence-based word dependency. All baselines are implemented carefully with the guidance of their official code.

Experiment Settings: In practice, we constructed document graphs with window size $s = 5$[2]. In particular, we mixed together with the same words in different positions as one node with word frequency to compute the correct likelihood. In other words, we implicitly assumed that the same words in one document share the topic assignment, which is widely used in [5, 30, 38, 46]. We set the Dirichlet prior parameter $\alpha = 1$. Following [42], batch-normalization [16] has been used on the inference network for the posterior parameters of θ. We utilized 300-dimensional GloVe word embeddings [37] to fix X (i.e., $L = 300$) in our model and word vectors in ETM. We also set the size of word vectors μ^v as 300, i.e., $H = 300$. The size of transitional vector a_k and b_k were set as $Y = 64$. For the optimization, We followed [30] and alternately updated the decoder parameters with topic representation and the encoder parameters. Only one sample is used in neural variational inference for θ_d and Z_d if needed. For the optimization, Adam [21] optimizer has been used with the initial learning rate of 0.001 and the linear learning rate decay trick to find the optimal. All quantitative results are based on 5 runs with different random seeds and the average performance is reported without special description.

4.2 Experimental Results

We followed [11] and selected the top 10 words with the highest probability in each topic as the representative word set to compute both TC and TD metrics. The results are displayed in Figure 3 with the varying number of topics K from 20 to 100 (see Appendix for more results based on the different number of top words). We find that our proposed model outperforms other baselines in terms of both C_v and NPMI score with high topic diversity on all datasets. In particular, GSM also achieves the competitive performance on TC metrics, but its TD score is lower than most models. It may be because that, in contrast to Dirichlet prior used in other models, the Gaussian-Softmax distribution in GSM leads to lower sparsity among topic proportions, which causes redundant topics in turn. Meanwhile, although some other baselines may have higher TD score than our model on several

[2] When $s = 5$, the degree of one word is 10, i.e., 5 in-degrees and 5 out-degrees, which is similar to that in Skip-gram with the size of text window as 5 that performs better in empirical experiments [32].

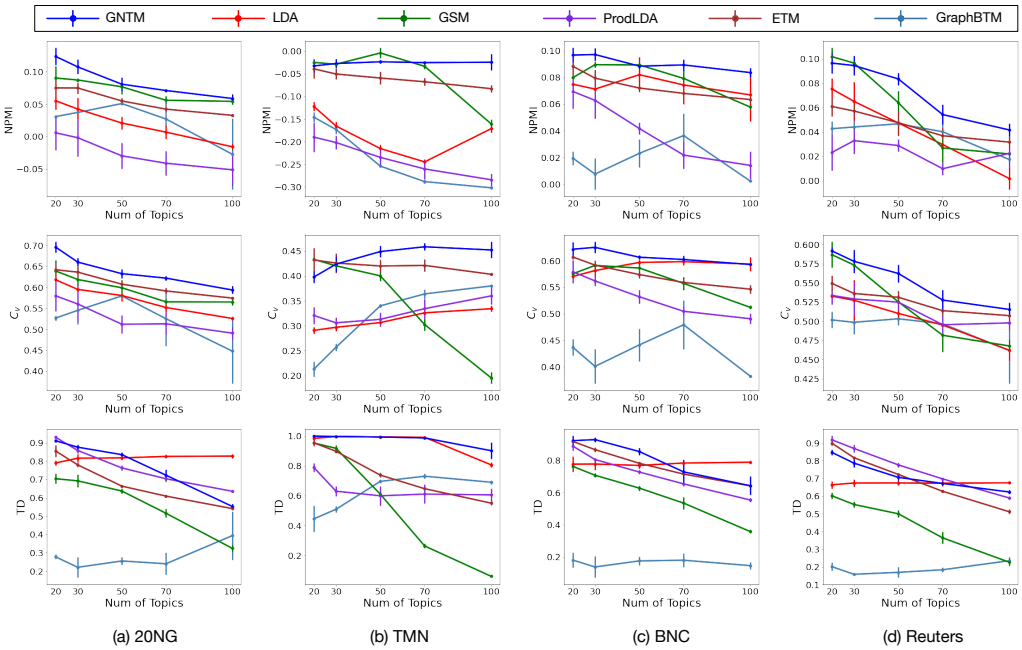


Figure 3: The performance on the topic quality. The first two rows show the topic coherence score, i.e., NPMI and C_v, with corresponding topic diversity in the third row, where error bars in each point denote the standard deviation of results on 5 runs. The result of iDcNADE has been omitted due to the uncompetitiveness. Completed results can be found in Appendix.

datasets with several topic numbers, they cannot achieve high topic coherence. In addition, despite considering word dependency, GraphBTM fails to capture the state-of-the-art performance. The reason may be that it construct each document graph on the whole vocabulary and include words that are not in the document as noise. In terms of document clustering with results in Table 1, our model performs the best with a significant gap over other models on Purity metric, while GNTM is also comparative on NMI metric. Those results demonstrates that our model captures both interpretable topics with better quality and good document representations for clustering. We also note that the GraphBTM fails to cluster documents. Especially, the NMI metric has vanished to 0 on the TMN dataset, which indicates that all documents were clustered to one category and GraphBTM cannot capture significant topics with similar semantics to the document labels.

Visualization: Since GNTM models topics as multinomial distributions on both the vocabulary and the word dependency edge sets, we can represent each topic k as a graph on the vocabulary V with the edge set E and adjacency matrix $A_k = (\beta_k^v \cdot (\beta_k^v)^T) \odot \beta_k^e$. We show two topics learned by a run of GNTM on News20 with $K = 20$ as examples in the left of Figure 4. First, we can find that the top words in each topic are highly semantically consistent, where topic #12 is about space science and topic #13 is related to computer technology. Second, we observe that the top edges are also

Table 1: The performance on document cluster. We highlight the best and second scores in boldface and with an underline, respectively. The standard deviation can be found in Appendix.

		Purity					NMI				
		20	30	50	70	100	20	30	50	70	100
20NG	LDA	0.2980	0.3340	0.3375	0.3510	0.3740	0.2908	0.3013	0.2852	0.2878	0.2858
	GSM	0.4133	0.4379	0.4629	0.4429	0.4210	0.4394	0.4369	**0.4433**	**0.4449**	**0.4412**
	ProdLDA	0.3306	0.3450	0.3641	0.3638	0.3807	0.3405	0.3345	0.3350	0.3298	0.3343
	ETM	0.3496	0.4154	0.4380	0.4510	0.4616	0.3842	0.4227	0.4296	0.4297	0.4356
	GraphBTM	0.1448	0.1210	0.1630	0.1068	0.0992	0.1552	0.1108	0.1807	0.0816	0.0707
	iDocNADE	0.2175	0.2844	0.3064	0.3187	0.3371	0.1128	0.1723	0.1802	0.1901	0.1990
	GNTM	**0.4500**	**0.4882**	**0.5089**	**0.5090**	**0.5021**	**0.4436**	**0.4419**	0.4416	0.4362	0.4371
TMN	LDA	0.3509	0.3692	0.3725	0.4031	0.4228	0.0622	0.0665	0.0754	0.0901	0.1064
	GSM	0.5933	0.6054	0.6184	0.5934	0.2632	0.2848	0.2787	**0.2996**	**0.3246**	0.0204
	ProdLDA	0.3141	0.2808	0.2505	0.2535	0.2438	0.0508	0.0334	0.0056	0.0053	0.0000
	ETM	0.5841	0.5967	0.6347	0.6358	0.6420	0.2764	0.2705	0.2829	0.2784	0.2767
	GraphBTM	0.2438	0.2438	0.2438	0.2438	0.2438	0.0000	0.0000	0.0000	0.0000	0.0000
	iDocNADE	0.2712	0.3116	0.3314	0.3512	0.3730	0.1074	0.1436	0.1488	0.1585	0.1701
	GNTM	**0.6150**	**0.6252**	**0.6472**	**0.6656**	**0.6773**	**0.2851**	**0.2801**	0.2789	0.2821	**0.2870**

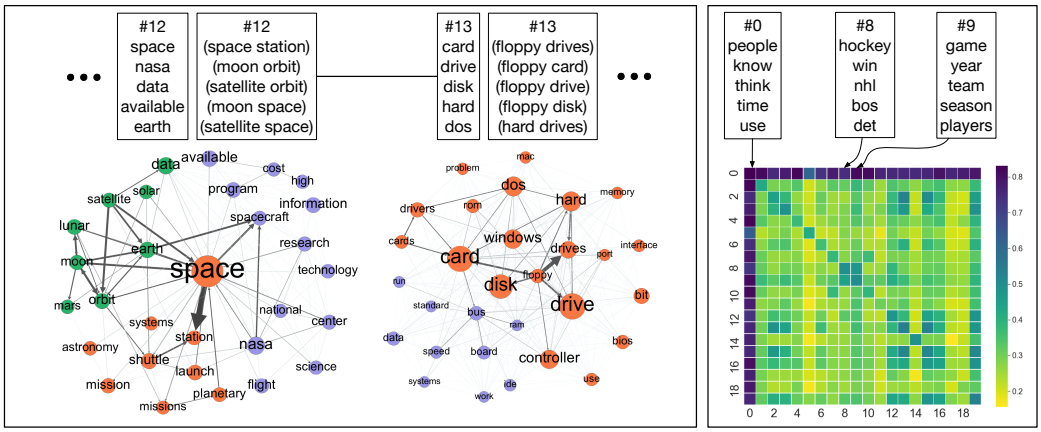


Figure 4: The case study. The left shows topics learned by GNTM with top 5 words, and top 5 edges and corresponding topic graph representation in the top 30 words, where the sizes of nodes and edges are positively correlated to β_k^v and $A_k = (\beta_k^v \cdot (\beta_k^v)^T) \odot \beta_k^e$, respectively. Nodes are colored by their cluster labels detected by fast unfloding algorithm [6]. The right displays the heatmap of the topic dependency matrix M, with several topics represented by the top 5 words.

highly interpretable with semantic dependency between two word nodes, such as "space station" and "satellite orbit" in topic #12, or "floppy dives" and "floppy disk' in topic #13, which also helps us to reduce the possible ambiguity and understand the specific meaning of words in each specific topic. Third, the graph representation of topics also provides a new possibility to further separate the topic semantic by clustering the word nodes. For example, we utilized the fast unfolding algorithm [6], a widely-used community detection algorithm, to cluster nodes and color different categories. We find that, interestingly, the top words of topic #12 are split into three sub-clusters with different sub-topics related to space research (violet), celestial bodies (green), and space exploration (orange), respectively. Similarly, the top words of the topic #13 are separated into two categories: one is about software or hardware (orange); the other may be about programming development (violet).

In addition, we also show the topic dependency matrix M in the right of Figure 4, where each element $m_{i,j}$ represents the possibility of existing a directed edge from topic i to j. We have the following interesting observations. First, there always exist high possibility values between topic #0 with any other topics. It is actually reasonable because topic #0 is represented with several general words, commonly used in any scenario. Second, the diagonal tend to have a high value. It indicates that the words with the same topic should be linked, which is consistent with the widely-used assumption that interdependent (or neighboring) words prefer to have the same topic assignment [13, 27, 28, 50, 51]. Third, we note that the topic dependency matrix also mines the dependency relation between different topics. For example, in our topic sets, the top words of the topic #9 is full of sport-related words, while the topic #8 focuses on the hockey games with keywords related to team or game name, such as "bos" (Boston Bruins), "det" (Detroit Red Wings) and "nhl" (National Hockey League). Considering the strong relation between topic #8 and #9, the value $m_{8,9}$ and $m_{9,8}$ is high.

5 Conclusion

In this paper, we revisited the topic modeling techniques by representing documents as semantic graphs and proposed a Graph Neural Topic Model (GNTM) method. Specifically, we first formulated a well-defined probabilistic generative story to model both the graph structure and word sets with a new concept of topics, i.e., multinomial distributions on both the vocabulary and word dependency edge. Then, an Autoencoder Variational Inference (AVI) approach was proposed to learn our model with graph neural networks to encode the document graphs. Besides, we also have demonstrated that LDA can be derived from GNTM as a special case with similar objective functions. Finally, extensive experiments on four benchmark datasets have clearly validated the effectiveness and interpretability of GNTM compared with state-of-the-art baselines.

Limitation and Future Work: Here, we discover word dependency only by linking words in a small sliding window on word sequence. Several other approaches can also construct document graphs, such as dependency parse [29]. An interesting direction of future work is to consider multiple types of word dependency relation together by representing documents as multi-relational graphs.

References

[1] Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 13–22, 2013.

[2] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

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

[4] David M Blei, John D Lafferty, et al. A correlated topic model of science. *The annals of applied statistics*, 1(1):17–35, 2007.

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

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[7] Chain Monte Carlo. Markov chain monte carlo and gibbs sampling. *Lecture notes for EEB*, 581, 2004.

[8] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175. Citeseer, 1994.

[9] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941, 2014.

[10] BNC Consortium et al. British national corpus. *Oxford Text Archive Core Collection*, 2007.

[11] Adji B Dieng, Francisco JR Ruiz, and David M Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 2020.

[12] 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.

[13] Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. Hidden topic markov models. In *Artificial intelligence and statistics*, pages 163–170. PMLR, 2007.

[14] Pankaj Gupta, Yatin Chaudhary, Florian Buettner, and Hinrich Schütze. Document informed neural autoregressive topic models with distributional prior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6505–6512, 2019.

[15] Philipp Hennig, David Stern, Ralf Herbrich, and Thore Graepel. Kernel topic models. In *Artificial intelligence and statistics*, pages 511–519. PMLR, 2012.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[17] Shoaib Jameel and Wai Lam. A nonparametric n-gram topic model with interpretable latent topics. In *Asia Information Retrieval Symposium*, pages 74–85. Springer, 2013.

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[19] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211, 2019.

[20] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[23] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[24] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995.

[25] Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. *Advances in Neural Information Processing Systems*, 25:2708–2716, 2012.

[26] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[27] Zhixing Li, Siqiang Wen, Juanzi Li, Peng Zhang, and Jie Tang. On modelling non-linear topical dependencies. In *International Conference on Machine Learning*, pages 458–466. PMLR, 2014.

[28] Robert Lindsey, William Headden, and Michael Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 214–222, 2012.

[29] Ryan McDonald and Joakim Nivre. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230, 2011.

[30] Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering discrete latent topics with neural variational inference. In *International Conference on Machine Learning*, pages 2410–2419. PMLR, 2017.

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

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

[33] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 262–272, 2011.

[34] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609, 2019.

[35] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108, 2010.

[36] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313, 2015.

[37] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[38] Mehdi Rezaee and Francis Ferraro. A discrete variational recurrent topic model without the reparametrization trick. *arXiv preprint arXiv:2010.12055*, 2020.

[39] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408, 2015.

[40] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. *arXiv preprint arXiv:1207.4169*, 2012.

[41] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[42] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

[43] Daniele Vitale, Paolo Ferragina, and Ugo Scaiella. Classification of short texts by deploying topical annotations. In *European Conference on Information Retrieval*, pages 376–387. Springer, 2012.

[44] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.

[45] Chong Wang and David M Blei. Variational inference in nonconjugate models. *Journal of Machine Learning Research*, 14(4), 2013.

[46] Tong Wang, Vish Viswanath, and Ping Chen. Extended topic model for word dependency. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 506–510, 2015.

[47] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 697–702. IEEE, 2007.

[48] Manzil Zaheer, Amr Ahmed, and Alexander J Smola. Latent lstm allocation: Joint clustering and non-linear dynamic modeling of sequence data. In *International Conference on Machine Learning*, pages 3967–3976. PMLR, 2017.

[49] He Zhao, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray Buntine. Topic modelling meets deep neural networks: A survey. *arXiv preprint arXiv:2103.00498*, 2021.

[50] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *European conference on information retrieval*, pages 338–349. Springer, 2011.

[51] Jun Zhu and Eric P Xing. Conditional topic random fields. 2010.

[52] Qile Zhu, Zheng Feng, and Xiaolin Li. Graphbtm: Graph enhanced autoencoded variational inference for biterm topic model. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4663–4672, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 4 and Appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1 and Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4.1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.1.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]