Neural Topic Models for Hierarchical Topic Detection and Visualization

Dang Pham [✉] and Tuan M. V. Le [✉]

Department of Computer Science
New Mexico State University, USA
{dangpnh,tuanle}@nmsu.edu

Abstract. Given a corpus of documents, hierarchical topic detection aims to learn a topic hierarchy where the topics are more general at high levels of the hierarchy and they become more specific toward the low levels. In this paper, we consider the joint problem of hierarchical topic detection and document visualization. We propose a joint neural topic model that can not only detect topic hierarchies but also generate a visualization of documents and their topic structure. By being able to view the topic hierarchy and see how documents are visually distributed across the hierarchy, we can quickly identify documents and topics of interest with desirable granularity. We conduct both quantitative and qualitative experiments on real-world large datasets. The results show that our method produces a better hierarchical visualization of topics and documents while achieving competitive performance in hierarchical topic detection, as compared to state-of-the-art baselines.

1 Introduction

Given a corpus of documents, hierarchical topic detection aims to learn a topic hierarchy where the topics are more general at high levels of the hierarchy and they become more specific toward the low levels. Flat topic models such as LDA [5] are not designed to detect topic hierarchies. Therefore, several hierarchical topic models including the nested Chinese restaurant process (nCRP) [4, 3], the nested hierarchical Dirichlet process (nHDP) [24] have been proposed to overcome this limitation. These models can learn the latent hierarchical structure of topics and they have a wide variety of applications such as language modeling [10], entity disambiguation [14], and sentiment analysis [1, 17]. More recently, there has been an increasing interest in neural approaches for topic modeling. Several flat neural topic models have been proposed for document modeling [7, 28], and supervised topic modeling [29]. For detecting topic hierarchies, we have neural methods such as TSNTM [11]. While traditional hierarchical topic models often use inference algorithms like collapsed Gibbs sampling or stochastic variational inference, TSNTM is trained using the autoencoding variational Bayes (AEVB) [18], which scales to large datasets.

Besides topic modeling, visualization is also an important tool for the analysis of text corpora. Topic modeling with visualization can provide users with an

effective overview of the text corpus which could help users discover useful insights without going through each document. Therefore, in this work, we investigate neural approaches for the joint problem of hierarchical topic detection and visualization. We propose a joint neural topic model that can not only detect topic hierarchies but also generate a visualization of documents and their topic structure. By being able to view the topic hierarchy as well as how documents are distributed across the hierarchy, users can quickly identify documents and topics of interest with desirable granularity. There are several types of visualization for visualizing topic hierarchies and documents including scatter plots [8], Sankey diagram [15], Sunburst diagram [26], and tag cloud [30]. In this work, we are interested in scatter plot visualization where documents, topics, and the topic hierarchy are embedded in a 2-d or 3-d visualization space. The joint problem of hierarchical topic detection and visualization can be formally stated as follows.

Problem. Let $\mathcal{D} = \{\mathbf{w}_n\}_{n=1}^{\mathcal{N}}$ denote a finite set of $\mathcal{N}$ documents and let $\mathcal{V}$ be a finite vocabulary from these documents. A document n is represented as a vector of word counts $\mathbf{w}_n \in \mathcal{R}^{|\mathcal{V}|}$. Given visualization dimension d: 1) For hierarchical topic modeling, we want to find a hierarchy structure of latent topics where each node in the hierarchy is a topic z and β_z is its word distribution. The hierarchy can have an infinite number of branches and nodes (topics). The most general topic is at the root node and more specific topics are at the leaf nodes. We also find topic distributions of documents that are collectively denoted as $\boldsymbol{\Theta} = \{\theta_n\}_{n=1}^{\mathcal{N}}$; 2) For visualization, we want to find d-dimensional visualization coordinates for $\mathcal{N}$ documents $\boldsymbol{X} = \{x_n\}_{n=1}^{\mathcal{N}}$, and all Z topics $\boldsymbol{\Phi} = \{\phi_z\}_{z=1}^{Z}$ such that the distances between documents, topics in the visualization space reflect the topic-document distributions $\boldsymbol{\Theta}$ as well as properties of the topic hierarchy.

There are three aspects considered in the stated problem. In the first aspect, we want to infer the latent topics in the text corpus. In the second aspect, we also want to organize these topics into a hierarchy. Finally, we want to visualize documents and their topics in the same visualization space for visual analysis. Most of the joint approaches so far only focus on one or two aspects. LDA [5] can learn topics but not their structure. nCRP [4], TSNTM [11] or other hierarchical topic models can both learn topics and organize them into a hierarchy. However, these topic models do not generate a visualization of documents and their topics. Therefore, recent topic models such as PLSV [12] and its variants [22, 21] are proposed to jointly infer topics and visualization using a single objective function. However, since they are flat topic models, they cannot learn or visualize the topic hierarchy.

In this paper, we aim to propose a neural hierarchical topic model, namely *HTV*, that jointly addresses all three aspects of the problem. In our approach, documents and topics are embedded in the same 2-d or 3-d visualization space. We introduce the path and level distributions over an infinite tree, and parameterize them by document and topic coordinates. To possibly create an unbounded topic tree, we use a doubly-recurrent neural network (DRNN) [2] to generate topic embeddings. Our contributions are as follows:

– We propose *HTV*, a novel visual hierarchical neural topic model for hierarchical topic detection and visualization.
– We develop an AEVB inference for our model that involves using a doubly-recurrent neural network (DRNN) over an infiniate tree and parameterizing the path and level distributions by document and topic coordinates. We also introduce the use of graph layout objective function of the Kamada-Kawai (KK) algorithm for visualizing the topic tree in our model.
– We conduct extensive experiments on several real-world datasets. The experimental results show that our method produces a better hierarchical visualization of topics and documents while achieving competitive performance in hierarchical topic detection, as compared to state-of-the-art baselines.

2 Visual and Hierarchical Neural Topic Model

2.1 Generative Model

In this section, we present the generative process of our proposed model. As shown in Figure 1, the topic hierarchy can be considered as a tree where each node is a topic. The tree could have an infinite number of branches and levels. The topic at the root is the most general and topics at the leaf nodes are more specific. To sample a topic for each word w_{nm} in a document n, a path c_{nm} from the root to a leaf node and a level l_{nm} are drawn. Let $\beta_{c_{nm}[l_{nm}]}$ be the topic distribution of the topic in the path c_{nm} and at level l_{nm}. The word w_{nm} is then drawn from the multinomial distribution Mult $\left(\beta_{c_{nm}[l_{nm}]}\right)$. The full generative process of *HTV* is as follows:

1. For each document $n = 1, \cdots, \mathcal{N}$:
   (a) Draw a document coordinate: $x_n \sim \text{Normal}\,(\mathbf{0}, \gamma \boldsymbol{I})$
   (b) Obtain a path distribution: $\pi_n = f_\pi(x_n, \boldsymbol{\Phi})$
   (c) Obtain a level distribution: $\delta_n = f_\delta(x_n, \boldsymbol{\Phi})$
   (d) For each word w_{nm} in document n:
      i. Draw a path: $c_{nm} \sim \text{Mult}\,(\pi_n)$
      ii. Draw a level: $l_{nm} \sim \text{Mult}\,(\delta_n)$
      iii. Draw a word: $w_{nm} \sim \text{Mult}\left(\beta_{c_{nm}[l_{nm}]}\right)$

Here $\boldsymbol{\Phi} = \{\phi_z\}_{z=1}^{Z}$ are coordinates of all topics in the tree, x_n is the coordinate of a document n. As in [11] [4], for each document n, besides topic distribution θ_n, we associate it with a path distribution π_n over all the paths from the root to the leaf nodes, and a level distribution δ_n over all tree levels. To possibly model the topic tree with an infinite number of branches and levels, nCRP [4] assumes that the level distribution is drawn from a stick-breaking construction:

$$\eta_l \sim \text{Beta}(1, \alpha), \delta_l = \eta_l \prod_{i=1}^{l-1}(1 - \eta_i), \tag{1}$$

and the path distribution is drawn from a nested stick-breaking construction as follows:

$$\upsilon_z \sim \text{Beta}(1, \varphi), \pi_z = \pi_{par(z)} \upsilon_z \prod_{z', z' \in ls(z)} (1 - \upsilon_{z'}) \tag{2}$$

here l is one of the levels, z is one of the topics in the topic tree, $par(z)$ is the parent topic of z, and $ls(z)$ represents the set of z's left siblings. η_l and υ_z are stick proportions of level l and topic z respectively. In our model, since we also want to visualize the topic tree, we need to formulate a way to encode these stick breaking constructions into the visualization space to make sure that the tree can grow unbounded. We introduce two functions $f_\pi(x_n, \boldsymbol{\Phi})$ and $f_\delta(x_n, \boldsymbol{\Phi})$ that are parameterized by document and topic visualization coordinates for computing the path distribution and the level distribution respectively.

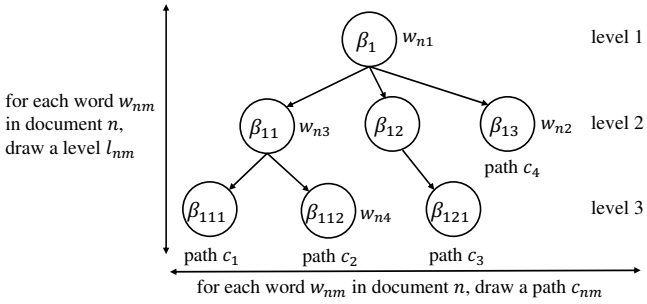


Fig. 1: Steps to sample a topic for each word w_{nm} in a document n. For each word w_{nm}, a path c_{nm} (from the root to a leaf node) and a level l_{nm} are sampled. The topic assigned to w_{nm} is $\beta_{c_{nm}[l_{nm}]}$. In this example, for the word w_{n3}, assume that the path c_1 and the level 2 are drawn. Topic β_{11} is then assigned to the word w_{n3}

2.2 Parameterizing Path Distribution and Level Distribution

In this section, we explain how path distribution and level distribution are parameterized by document and topic visualization coordinates. From Eq. 2, generally for all topics that are children of a parent node p, this will hold:

$$\sum_{z, z \in child(p)} \pi_z = \pi_p \Longleftrightarrow \sum_{z, z \in child(p)} \frac{\pi_z}{\pi_p} = 1 \tag{3}$$

here $child(p)$ represents the set of all children of the parent node p. Let $\tau_z = \frac{\pi_z}{\pi_p} = \frac{\pi_z}{\pi_{par(z)}}$. To encode the nested stick breaking construction of the path distribution into the visualization, we parameterize τ_z of each document n as a function of the distance between x_n and ϕ_z as follows:

$$\tau_{zn} = \frac{\rho(\|x_n - \phi_z\|)}{\sum_{z', z' \in child(par(z))} \rho(\|x_n - \phi_{z'}\|)} \tag{4}$$

here $child(par(z))$ represents the set of all children of parent of z, the denominator is for normalization so that (3) still holds, and ρ is a radial basis function (RBF) which can have different forms such as Gaussian: $\exp(-\frac{1}{2}r^2)$, or Inverse quadratic: $\frac{1}{1+r^2}$ where $r = \|x - \phi_z\|$ is the distance from x_n to topic coordinate ϕ_z [1]. Eq. 4 with Gaussian ρ is also used in PLSV to encode the topic distribution in the visualization space [12]. As shown in [25], Inverse quadratic consistently produces good performance and in some cases it gives better results. Therefore, we choose to use Inverse quadratic in our experiments. Eq. 4 becomes:

$$\tau_{zn} = \frac{\frac{1}{1+\|x_n-\phi_z\|^2}}{\sum_{z',z'\in child(par(z))} \frac{1}{1+\|x_n-\phi_{z'}\|^2}} \tag{5}$$

As we can see from the above formula, when the document n is close to the topic z in the visualization space, the numerator will be high and thus τ_{zn} and $\pi_{zn} = \tau_{zn}\pi_{par(z)n}$ will be high. Therefore, in step (1)(d)(i) of the generative process, the words in document n tend to be assigned to the paths that going through topic z.

Note that π_n is the path distribution of a document n. It is easy to see that the number of paths is equal to the number of leaf nodes in the topic tree. Therefore, π_{in} of the leaf node i is the path proportion of the path that goes to the leaf node i and it is computed as follows:

$$\pi_{in} = \tau_{in}\pi_{par(i)n} = \prod_{z,z\in path(i)} \tau_{zn} \tag{6}$$

here note that $\pi_{root} = 1$ and $path(i)$ represents all the nodes that lie on the path from the root to the leaf node i. From (6), π_n is then a function of $x_n, \boldsymbol{\Phi}$, i.e., $\pi_n = f_\pi(x_n, \boldsymbol{\Phi})$, which is used in step (1)(b) of the generative process.

Similarly, we also parameterize the level distribution of a document n as a function of x_n and topic coordinates $\boldsymbol{\Phi}$:

$$\delta_{ln} = \frac{\frac{1}{1+\min\{\|x_n-\phi_z\|^2,\forall z \text{ in level } l\}}}{\sum_{l'=1}^{L} \frac{1}{1+\min\{\|x_n-\phi_{z'}\|^2,\forall z' \text{ in level } l'\}}} \tag{7}$$

where $\min\{\|x_n - \phi_z\|^2, \forall z \text{ in level } l\}$ is the minimum distance between a document n and all topics in the l-th level. From (7), δ_n is a function of $x_n, \boldsymbol{\Phi}$, i.e., $\delta_n = f_\delta(x_n, \boldsymbol{\Phi})$, which is used in step (1)(c) of the generative process. Based on π_n and δ_n, the topic distribution θ_n can be derived as: $\theta_{zn} = (1 - \sum_{l=1,l\neq l_z}^{L} \delta_{ln})(\sum_{c:c_l=z} \pi_{cn})$, where l_z is the level of topic z.

2.3 Parameterizing Word Distribution

Let $t_z \in \mathbb{R}^H$ be the embedding of topic z and $U \in \mathbb{R}^{V\times H}$ be the embeddings of words. The word distribution of topic z is computed as: $\beta_z = \text{softmax}(\frac{U\cdot t_z^T}{\kappa^{\frac{1}{l_z}}})$,

[1] r is Euclidean distance in our experiments

where $\kappa^{\frac{1}{l_z}}$ is the temperature value that controls the sparsity of β_z. When the level l_z is deeper, the probability distribution over words β_z is sparser [11]. To possibly create an unbounded topic tree, as in [11] we use a doubly-recurrent neural network (DRNN) [2] to generate topic embeddings. A DRNN consists of two RNNs that respectively model the ancestral (parent-to-children) and fraternal (sibling-to-sibling) flows of information in the topic tree. More specifically, the hidden state h_z of the topic z is given by:

$$h_z = \tanh(W_h(\tanh(W_p h_{par(z)} + b_p) + \tanh(W_s h_{z-1} + b_s)) + b_h) \tag{8}$$

where $\tanh(W_p h_{par(z)} + b_p)$ and $\tanh(W_s h_{z-1} + b_s)$ can be considered as the ancestral and fraternal hidden states. The output topic embedding t_z is computed based on h_z as: $t_z = W h_z + b$. To increase the diversity of topics in the tree while allowing parent-children correlations, as in [11] we apply the following tree-specific diversity regularizer to the final objective function (Section 2.6):

$$L_{td} = \sum_{z \notin Leaf} \sum_{i,j \in Child(z); i \neq j} \left(\frac{\bar{t}_{zi}^{\top} \cdot \bar{t}_{zj}}{\left\|\bar{t}_{zi}\right\| \left\|\bar{t}_{zj}\right\|} - 1 \right)^2 \tag{9}$$

where $\bar{t}_{zi} = t_i - t_z$, $Leaf$ and $Child(z)$ denote the set of the topics with no children and the children of the z topic, respectively.

2.4 Visualizing the Topic Tree

Our model also aims to visualize the topic tree. While the model can learn the topic visualization coordinates, it does not guarantee that the edges connecting topics do not cross each other. Therefore, to ensure that we have a visually appealing layout of the topic tree (e.g., the number of crossing edges is minimized), we employ the graph layout objective function of the Kamada-Kawai (KK) algorithm [13] and use it to regularize the topic coordinates in our final objective function (Section 2.6). The layout objective function of the KK algorithm is specified as: $L_{kk} = \sum_{i \neq j} \frac{1}{2}(\frac{d_{i,j}}{s_{i,j}} - 1)^2$, where, in our case, $d_{i,j} = ||\phi_i - \phi_j||$ is the Euclidean distance between topics i,j in the visualization space, $s_{i,j}$ is the graph-theoretic distance of the shortest-path between topics i, j in the tree. The weight of the edge connecting topics i, j is computed based on the cosine distance between topic embeddings t_i, t_j, i.e., $\text{weight}(i, j) = 1 + \text{cosine\_dist}(t_i, t_j)$. Intuitively, two connected topics that are not similar will result in a longer edge.

2.5 Dynamically Growing the Topic Tree

We explain how the model dynamically updates the tree using heuristics. For each topic z, we estimate the proportion of the words in the corpus belonging to topic z: $p_z = \frac{\sum_{n=1}^{\mathcal{N}} M_n \hat{\theta}_{zn}}{\sum_{n=1}^{\mathcal{N}} M_n}$, where M_n is the number of words in document n. We compare p_z with the level-dependent pruning and adding thresholds to determine whether z should be removed or a new child topic should be added to

z for refining it. We use the adding threshold defined as $\max(a, \frac{1}{min\_child*2^{l-1}})$ and the pruning threshold defined as $\max(b, \frac{1}{max\_child*2^{l-1}})$. Here, min_child and max_child can be interpreted as the expected minimum and maximum numbers of children. The max function is to ensure that the thresholds are not too small when the number of levels is increasing [2]. For a topic z that has p_z greater than the adding threshold: 1) If z is a non-leaf node, one child is added to z; (2) If z is a leaf node, two children are added. This is the case where the model grows the tree by increasing the number of levels. Finally, if the sum of the proportions of all descendants of topic z, i.e., $\sum_{j\in \mathrm{Des}(z)} p_j$ is smaller than the pruning threshold then z and its descendants are removed.

2.6 Autoencoding Variational Inference

In this section, we present the inference of our model based on AEVB. The marginal likelihood of a document is given by:

$$\begin{aligned} p(\mathbf{w}_n|\boldsymbol{\Phi},\boldsymbol{\beta},\gamma) &= \int_x \Big\{ \prod_m \sum_{c,l} p(w_{nm}|\boldsymbol{\beta}_{c[l]}) p(c|x,\Phi) p(l|x,\boldsymbol{\Phi}) \Big\} p(x|\gamma) dx \\ &= \int_x \Big\{ \prod_m \sum_z p(w_{nm}|\boldsymbol{\beta}_z)\theta_{zn} \Big\} p(x|\gamma) dx \end{aligned} \tag{10}$$

where $p(c|x,\boldsymbol{\Phi})$, $p(l|x,\boldsymbol{\Phi})$ are the path distribution π_n and the level distribution δ_n respectively. They are computed as in Eq. 6 and Eq. 7. θ_n is the topic distribution and $\theta_{zn} = (1 - \sum_{l=1,l\neq l_z}^{L} \delta_{ln})(\sum_{c:c_l=z} \pi_{cn})$. Based on the AEVB framework, we have the following lower bound to the marginal log likelihood (ELBO) of a document:

$$\mathcal{L}\left(\eta|\gamma,\boldsymbol{\Phi},\boldsymbol{\beta}\right) = -\mathbb{D}_{\mathrm{KL}}\left[q(x|\mathbf{w}_n,\eta)\|p(x|\gamma)\right] + \mathbb{E}_{q(x|\mathbf{w}_n,\eta)}\left[\log\left(\theta_n\boldsymbol{\beta}\right)\mathbf{w}_n^T\right] \tag{11}$$

where $q(x|\mathbf{w}_n,\eta) = \mathrm{Normal}\left(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\right)$ is the variational distribution and $\boldsymbol{\mu}_n$, diagonal $\boldsymbol{\Sigma}_n \in \mathbb{R}^d$ are outputs of the encoding feed forward neural network with variational parameters η. The whole inference network architecture including the DRNN of HTV is shown in Figure 8. To estimate the expectation w.r.t $q(x|\mathbf{w}_n,\eta)$ in Eq. 11, we sample an $\hat{x}$ from the posterior $q(x|\mathbf{w}_n,\eta)$ by using reparameterization trick, i.e., $\hat{x} = \boldsymbol{\mu}_n + \boldsymbol{\Sigma}_n^{1/2}\hat{\boldsymbol{\epsilon}}$ where $\hat{\boldsymbol{\epsilon}} \sim \mathrm{Normal}\left(\mathbf{0},\boldsymbol{I}\right)$ [18]. For the whole corpus, the lower bound is then approximated as:

$$\begin{aligned} \mathcal{L}(\Omega) = \sum_{n=1}^{\mathcal{N}} \Bigg[&-\frac{1}{2}\left(\mathrm{tr}\left((\gamma\boldsymbol{I})^{-1}\boldsymbol{\Sigma}_n\right) + \left(-\boldsymbol{\mu}_n\right)^T(\gamma\boldsymbol{I})^{-1}\left(-\boldsymbol{\mu}_n\right) - d + \log\frac{|\gamma\boldsymbol{I}|}{|\boldsymbol{\Sigma}_n|}\right) \\ &+ \log\left(\hat{\theta}_n\boldsymbol{\beta}\right)\mathbf{w}_n^T\Bigg] \end{aligned} \tag{12}$$

Adding the tree-specific diversity regularizer (Eq. 9) and the KK layout regularizer, we have the final objective function:

$$\mathcal{L} = \mathcal{L}(\Omega) + \lambda_{td} * L_{td} + \lambda_{kk} * L_{kk} \tag{13}$$

[2] In the experiments, we set $a = b = 0.01$

3 Experiments

Datasets. In our experiments, we use four real-world datasets: 1) BBC[3] consists of 2225 documents from BBC News [9]. It has 5 classes: business, entertainment, politics, sport, and tech; 2) Reuters[4] contains 7674 newswire articles from 8 categories [6]; 3) 20 Newsgroups[5] contains 18251 newsgroups posts from 20 categories; 4) Web of Science[6] contains the abstracts and keywords of 46,985 published papers from 7 research domains: CS, Psychology, Medical, ECE, Civil, MAE, and Biochemistry [19]. All datasets are preprocessed by stemming and removing stopwords. The vocabulary sizes are 2000, 3000, 3000, and 5000 for BBC, Reuters, 20 Newsgroups, and Web of Science respectively.
Comparative Baselines. We compare our proposed model with the following baselines: 1) *LDA-VAE*[7]: LDA with variational auto-encoder (VAE) inference [27]; 2) *PLSV-VAE*[8]: PLSV using VAE inference with Inverse quadratic RBF [25]; 3) *nCRP*[9]: A hierarchical topic model based on the nested Chinese restaurant process with collapsed Gibbs sampling [3]; 4) *TSNTM*[10]: A hierarchical neural topic model using VAE inference [11]; 5) *HTV* (our model): A novel joint model for both hierarchical topic modeling and visualization with VAE inference.

LDA-VAE, *nCRP*, and *TSNTM* are methods for topic modeling but they do not produce visualization. Therefore, for these methods, we use t-SNE [23] [11] to embed the documents' topic proportions for visualization. In contrast, *PLSV-VAE* and *HTV* are joint methods for both topic modeling and visualization. Although *PLSV-VAE* is a flat topic model that does not detect topic hierarchies, for completeness we will compare our method with it. In our experiments, VAE-based methods are trained by AdaGrad with 2000 epochs, learning rate 0.01, batch size 512, and dropout with probability $p = 0.2$. For *TSNTM* and *HTV*, we use 256-dimensional word and topic embeddings, and $\kappa = 0.1$ for computing temperature value in β_z. The adding and pruning thresholds of *TSNTM* are 0.01 and 0.005 respectively. In *HTV*, we experimentally set (*min_child*, *max_child*) as (5, 10) for BBC, (5, 15) for Reuters, and (6, 15) for both 20 Newsgroups and Web of Science. These values work well for these datasets. We set the regularization parameters as $\lambda_{td} = 0.1$ and $\lambda_{kk} = 1000$, which consistently produces good performance across all datasets. Smaller λ_{kk} would result in more crossing edges. We initialize the tree with 3 levels where each node has 3 children. The maximum level is set to 4.

Different from hierarchical methods, *PLSV-VAE* and *LDA-VAE* need the number of topics to be specified before training. For a fair comparison, we set the

[3] http://mlg.ucd.ie/datasets/bbc.html

[4] http://ana.cachopo.org/datasets-for-single-label-text-categorization

[5] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

[6] https://data.mendeley.com/datasets/9rw3vkcfy4/6

[7] Its implementation is at https://github.com/akashgit/autoencoding_vi_for_topic_models

[8] We use the implementation at https://github.com/dangpnh2/plsv_vae

[9] We use the implementation at https://github.com/blei-lab/hlda

[10] We use the implementation at https://github.com/misonuma/tsntm

[11] https://github.com/DmitryUlyanov/Multicore-TSNE

number of topics in *PLSV-VAE* and *LDA-VAE* to be equal to the number of topics generated by *HTV* for each run. Regarding *nCRP*, we set its hyperparameters as follows: $\gamma = 0.01$, the Dirichlet parameter $\eta = 5$, and the GEM parameters are set as $\pi = 10$ and $m = 0.5$. All the experiment results are averaged across 10 runs on a system with 64GB memory, an Intel(R) Xeon(R) CPU E5-2623v3, 16 cores at 3.00GHz. The GPU in use on this system is NVIDIA Quadro P2000 GPU with 1024 CUDA cores and 5 GB GDDR5.

3.1 Tree-Structure and Visualization Quantitative Evaluation

We evaluate the quality of the tree structure using document specialization in the visualization space and two other metrics: node specialization and hierarchical affinity that are also used in [11][16].

Document Specialization in the Visualization Space. In this task, we measure the quality of hierarchical visualization of documents and topics. A good hierarchical visualization should put general documents close to general topics and the farther the documents are from the root, the more specific they are. We quantify this aspect by finding the top $5\%, 10\%, \ldots$ of all documents that are the closest to the root topic in the visualization space. For each such set of documents, we compute the average cosine similarity between each document and the vector of the entire corpus. As in [11][16], the vector of the entire corpus is computed based on the frequencies of the words and is considered as the most general topic. We would expect that the average cosine similarity will be high for documents near the root and it will be decreasing when farther away. Since *PLSV-VAE*, *LDA-VAE*, *nCRP*, and *TSNTM* do not visualize topics, we use the average of all documents coordinates as the root. Figure 2 shows the average cosine similarity (i.e., doc specialization as in the figure) by the methods for different top $k\%$ of documents. The high steepness of the curve by our model *HTV* indicates that the documents are organized better into hierarchies in the visualization where the most general documents are near the root and they become increasingly specific when farther away.

Classification in Visualization Space. We show that while producing better hierarchical visualization, our method still generates a high quality scatterplot visualization in terms of k-NN accuracy in the visualization space. k-NN accuracy is widely used to evaluate the quality of the visualization [23][25]. In this evaluation approach, a k-NN classifier is used to classify documents using their visualization coordinates. A good visualization should group documents with the same label together and hence yield a high classification accuracy in the visualization space. Figure 3 shows k-NN accuracy of all models across datasets. This figure shows that visualization by *HTV* is as good as other methods. This will be further confirmed when we look at the example visualizations in Section 3.3.

Node Specialization. A good tree structure should have the general topics near the root and topics become more specific toward the low levels. To quantify this aspect, we rely on node specialization that measures the specialization score as the cosine distance between the word distribution of each topic and the vector of the entire corpus [11]. Since the entire corpus vector is regarded as the most

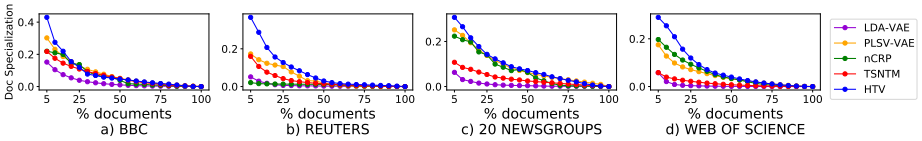


Fig. 2: Document specialization

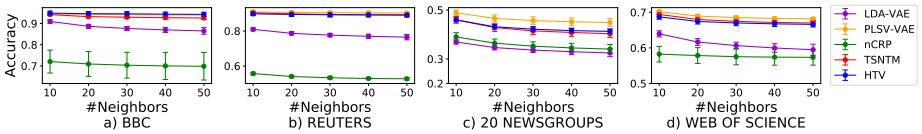


Fig. 3: k-NN accuracy in the visualization space

general topic, more specific topics should have higher cosine distances. Table 1 shows the average cosine distance of all topics at each level. We only compare our model with hierarchical methods in this task. Except for *TSNTM* in BBC, the specialization scores for each model increase as the level increases.

Hierarchical Affinity. Another characteristic of a good tree structure is that a parent topic should be more similar to its children than the topics descended from the other parents. As in [11], we compute the average cosine similarity between a node to its children and non-children nodes. Table 2 shows the average cosine similarity over the topics of all models. The higher score over child nodes indicates that a parent is more similar to its child nodes. We only show the results of hierarchical methods in this task. All three models infer child topics similar to their parents.

3.2 Topic Coherence and Running Time Comparison

We evaluate the quality of topic models produced by all methods in terms of topic coherence. The objective is to show that while generating better hierarchical visualization quality, *HTV* also achieves competitive performance on topic coherence. For topic coherence, we use the Normalized Pointwise Mutual Information (NPMI) [20] estimated based on a large external corpus. We use Wikipedia 7-gram dataset created from the Wikipedia dump data as of June 2008 version [12]. Table 3 shows the average Normalized Pointwise Mutual Information (NPMI [20]) over all topics for all models. The NPMI scores of *HTV* over all datasets are comparable to all baselines. Comparing to hierarchical methods *nCRP* and *TSNTM*, *HTV* can find slightly better topics. For running time, since *HTV* uses VAE inference, it scales well to large datasets. As shown in Table 4, it runs much faster than *nCRP* and has comparable running time to *TSNTM*.

3.3 Visualization Qualitative Evaluation

Figures 4, 5, and 6 show visualization examples by *HTV*, *PLSV-VAE*, and *TSNTM* on Reuters, 20 Newsgroups, and Web of Science respectively. Each colored point represents a document, and the larger points with black

[12] https://nlp.cs.nyu.edu/wikipedia-data/

Table 1: Topic specialization scores. Except *TSNTM* method in BBC from level 3 to level 4, the scores increase as the level increases for all models

Dataset	Model	Level 1		Level 2		Level 3		Level 4
BBC	*nCRP*	0.188	<	0.529	<	0.792	<	0.845
	TSNTM	0.321	<	0.528	<	0.557	>	0.516
	HTV	0.339	<	0.579	<	0.722	<	0.831
Reuters	*nCRP*	0.097	<	0.612	<	0.815	<	0.882
	TSNTM	0.315	<	0.535	<	0.563	<	0.566
	HTV	0.450	<	0.561	<	0.739	<	0.877
20 Newsgroups	*nCRP*	0.097	<	0.612	<	0.847	<	0.894
	TSNTM	0.247	<	0.456	<	0.538	<	0.561
	HTV	0.447	<	0.452	<	0.672	<	0.802
Web of Science	*nCRP*	0.148	<	0.606	<	0.814	<	0.870
	TSNTM	0.306	<	0.439	<	0.511	<	0.518
	HTV	0.411	<	0.431	<	0.671	<	0.754

Table 2: Hierarchical Affinity. Except *TSNTM* method in BBC from level 3 to level 4, the scores increase as the level increases for all models

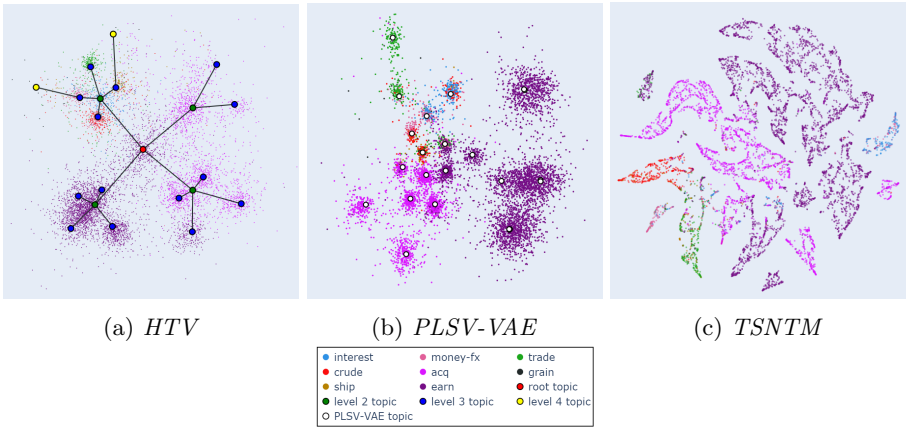
Dataset	Model	Child	Non-Child
BBC	*nCRP*	0.146	0.063
	TSNTM	0.201	0.171
	HTV	0.127	0.060
Reuters	*nCRP*	0.139	0.095
	TSNTM	0.254	0.188
	HTV	0.151	0.070
20 Newsgroups	*nCRP*	0.138	0.095
	TSNTM	0.238	0.194
	HTV	0.146	0.081
Web of Science	*nCRP*	0.140	0.089
	TSNTM	0.275	0.205
	HTV	0.143	0.081

Table 3: Average NPMI of all topics over 10 runs

model	BBC	Reuters	20 Newsgroups	Web of Science
LDA	0.091	0.051	0.95	0.094
PLSV-VAE	0.095	0.054	0.095	0.099
nCRP	0.043	0.039	0.031	0.053
TSNTM	0.090	0.053	0.092	0.094
HTV(Our model)	0.091	0.052	0.094	0.099

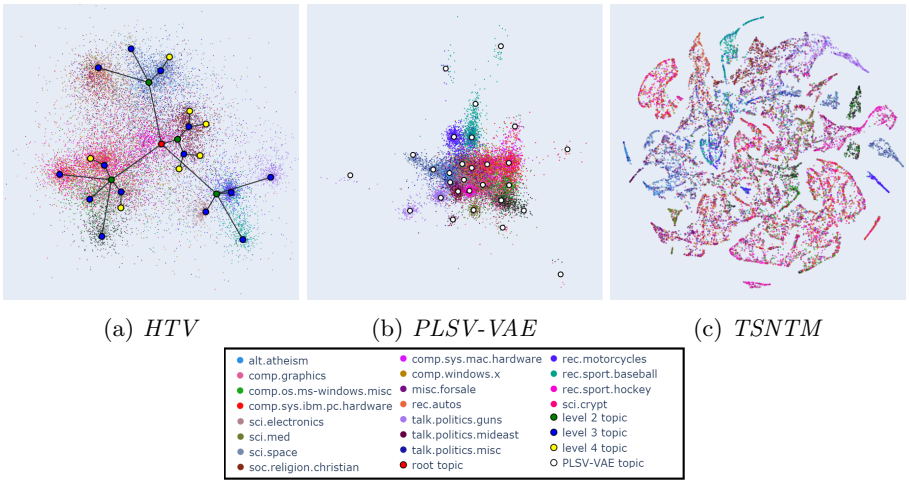
Table 4: Running time (in seconds) of three models: *nCRP*, *TSNTM*, and *HTV*

Dataset	*nCRP*	*TSNTM*	*HTV*
BBC	84120	6008	7223
Reuters	31300	2132	2247
20 Newsgroups	7079	1011	882
Web of Science	5535	294	295



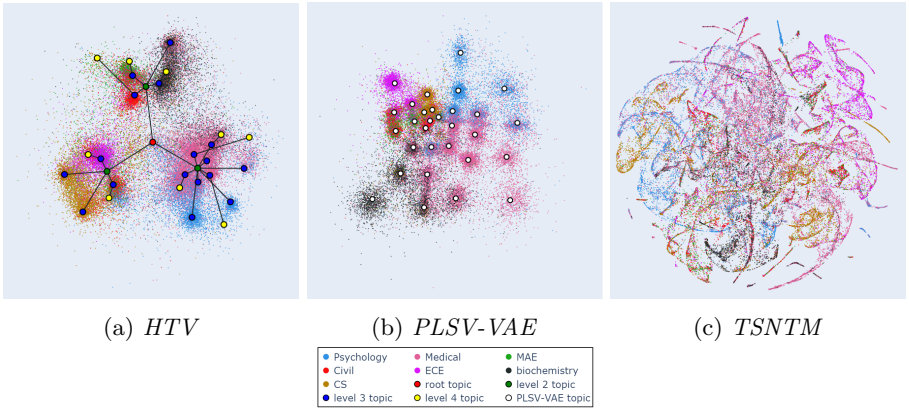
(a) *HTV* (b) *PLSV-VAE* (c) *TSNTM*

Fig. 4: Visualization of Reuters by a) *HTV* b) *PLSV-VAE* c) *TSNTM*



(a) *HTV* (b) *PLSV-VAE* (c) *TSNTM*

Fig. 5: Visualization of 20 Newsgroups by a) *HTV* b) *PLSV-VAE* c) *TSNTM*



(a) *HTV* (b) *PLSV-VAE* (c) *TSNTM*

Fig. 6: Visualization of Web of Science by a) *HTV* b) *PLSV-VAE* c) *TSNTM*

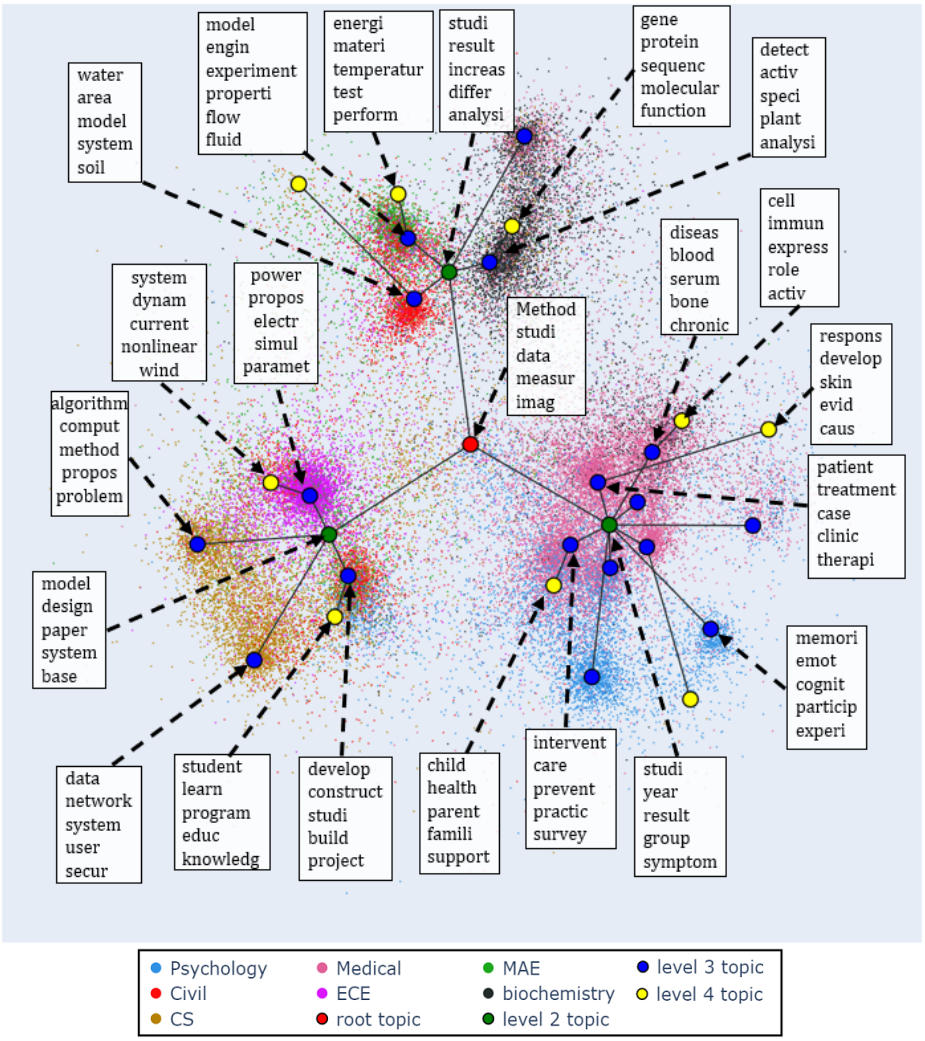


Fig. 7: Visualization and hierarchical topics found by *HTV* on Web of Science

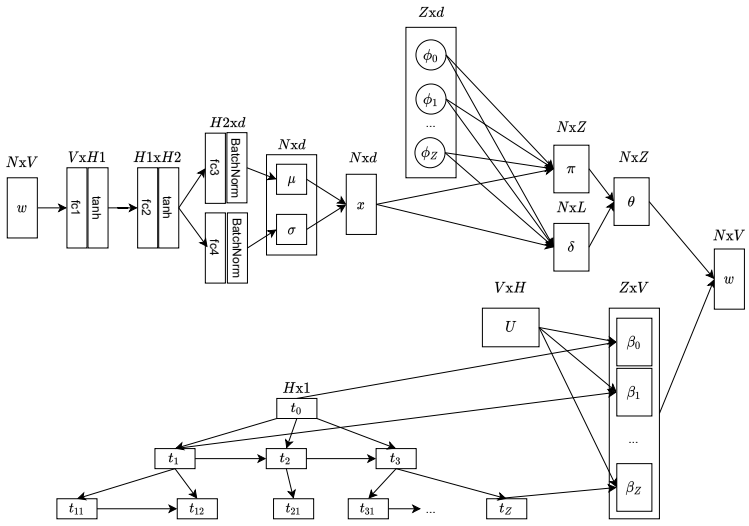


Fig. 8: The inference network architecture of *HTV*

border are topics (only in *PLSV-VAE* and *HTV*). In *HTV*, the red point with black border is the root topic and the green points with black border are the level 2 topics, finally, the blue and yellow points with black border represent level 3 and level 4 topics. It is clear that *HTV* can find good document clusters as compared to *PLSV-VAE* and *TSNTM* with t-SNE. Moreover, *HTV* learns a topic tree for each dataset and visualizes it in the visualization space using KK layout objective function. This helps to minimize the crossing edges as seen in all the visualization examples. In contrast, *TSNTM* does not visualize topics, and for *PLSV-VAE*, it does not infer the topic hierarchy. Therefore, it is difficult to tell the relationship between topics in the visualization. In Figure 7, we show the visualization of documents along with the generated topics by *HTV* on Web of Science. The inferred topic tree has three branches. The root topic has top 5 words: "method, studi, data, measur, img" which are very general words in sciences domain. As we can see, topics at the lower levels are more specific. For example, topics on levels 3, 4 in the top branch are very specific. They are topics in Civil, MAE, Biochemistry domains such as "water, are, model, system, soil", "model, engin, experiment, properti, flow, fluid", and "gene, protein, sequenc, molecular, function". The layout of topics and their structure show that our model can extract the topic hierarchy and visualize it along with the documents.

4 Related Work

Hierarchical structure is an effective way to organize topics as it helps users to understand and explore the structure of topics. Flat topic models such as LDA [5] are not designed to detect topic hierarchies. Therefore, several hierarchical topic models including the nested Chinese restaurant process (nCRP) [4, 3], the nested hierarchical Dirichlet process (nHDP) [24] have been proposed to overcome this limitation. Recently, there has been an increasing interest in neural approaches for topic modeling [7, 28, 29]. For detecting topic hierarchies, we have neural methods such as TSNTM [11], which is a neural extension of nCRP. TSNTM parameterizes the topic distribution over an infinite tree by a doubly-recurrent neural network (DRNN). TSNTM is trained using AEVB, making it scale well to larger datasets than the nCRP-based model.

All of the above methods work well for topic modeling but they are not designed for visualization tasks. Therefore, several works including the pioneering model PLSV [12] and its variants [22] [21] have been proposed to jointly perform topic modeling and visualization. PLSV is a flat topic model where a generative model is used to generate both topics and visualization. Recently, *PLSV-VAE* [25] proposes using AEVB for scalable inference in PLSV. These joint models are not for hierarchical topic detection. To the best of our knowledge, our model is the first joint model for detecting topic hierarchies and visualization.

5 Conclusion

In this paper, we propose *HTV*, a visual hierarchical neural topic model for jointly detecting topic hierarchies and visualization. We parameterize the path

distribution and level distribution by document and topic coordinates. To possibly create an unbounded topic tree, we use a DRNN to generate topic embeddings. We make use of KK layout objective function to regularize the model, ensuring that we have a visually appealing layout of the topic tree in the visualization space. Our extensive experiments on four real-world datasets show that *HTV* generates better hierarchical visualization of documents and topics while gaining competitive performance in hierarchical topic detection, as compared to state-of-the-art baselines.

Acknowledgments

This research is sponsored by NSF #1757207 and NSF #1914635.

References

1. Almars, A., Li, X., Zhao, X.: Modelling user attitudes using hierarchical sentiment-topic model. Data & Knowledge Engineering **119**, 139–149 (2019)
2. Alvarez-Melis, D., Jaakkola, T.: Tree-structured decoding with doubly-recurrent neural networks. In: ICLR (2017)
3. Blei, D.M., Griffiths, T.L., Jordan, M.I.: The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. Journal of the ACM (JACM) **57**(2), 1–30 (2010)
4. Blei, D.M., Griffiths, T.L., Jordan, M.I., Tenenbaum, J.B.: Hierarchical topic models and the nested chinese restaurant process. Advances in neural information processing systems **16**(16), 17–24 (2004)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research **3**, 993–1022 (2003)
6. Cardoso-Cachopo, A.: Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa (2007)
7. Chen, Y., Zaki, M.J.: Kate: K-competitive autoencoder for text. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 85–94 (2017)
8. Choo, J., Lee, C., Reddy, C.K., Park, H.: Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. IEEE transactions on visualization and computer graphics **19**(12), 1992–2001 (2013)
9. Greene, D., Cunningham, P.: Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proc. 23rd International Conference on Machine learning (ICML'06). pp. 377–384. ACM Press (2006)
10. Guo, D., Chen, B., Lu, R., Zhou, M.: Recurrent hierarchical topic-guided RNN for language generation. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 3810–3821 (2020)
11. Isonuma, M., Mori, J., Bollegala, D., Sakata, I.: Tree-structured neural topic model. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 800–806 (2020)
12. Iwata, T., Yamada, T., Ueda, N.: Probabilistic latent semantic visualization: topic model for visualizing documents. In: KDD. pp. 363–371 (2008)
13. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Information Processing Letters (1989)

14. Kataria, S.S., Kumar, K.S., Rastogi, R.R., Sen, P., Sengamedu, S.H.: Entity disambiguation with hierarchical topic models. In: KDD. pp. 1037–1045 (2011)
15. Kim, H., Drake, B., Endert, A., Park, H.: Architext: Interactive hierarchical topic modeling. IEEE transactions on visualization and computer graphics (2020)
16. Kim, J.H., Kim, D., Kim, S., Oh, A.H.: Modeling topic hierarchies with the recursive chinese restaurant process. Proceedings of the 21st ACM international conference on Information and knowledge management (2012)
17. Kim, S., Zhang, J., Chen, Z., Oh, A., Liu, S.: A hierarchical aspect-sentiment model for online reviews. In: AAAI. vol. 27 (2013)
18. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
19. Kowsari, K., Brown, D.E., Heidarysafa, M., Jafari Meimandi, K., , Gerber, M.S., Barnes, L.E.: Hdltex: Hierarchical deep learning for text classification. In: International Conference on Machine Learning and Applications. IEEE (2017)
20. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 530–539 (2014)
21. Le, T., Lauw, H.: Manifold learning for jointly modeling topic and visualization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 28 (2014)
22. Le, T.M., Lauw, H.W.: Semantic visualization for spherical representation. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1007–1016 (2014)
23. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
24. Paisley, J., Wang, C., Blei, D.M., Jordan, M.I.: Nested hierarchical dirichlet processes. IEEE transactions on pattern analysis and machine intelligence **37**(2), 256–270 (2014)
25. Pham, D., Le, T.: Auto-encoding variational bayes for inferring topics and visualization. In: Proceedings of the 28th International Conference on Computational Linguistics (2020)
26. Smith, A., Hawes, T., Myers, M.: Hiearchie: Visualization for hierarchical topic models. In: Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces. pp. 71–78 (2014)
27. Srivastava, A., Sutton, C.A.: Autoencoding variational inference for topic models. In: ICLR (2017)
28. Wang, R., Hu, X., Zhou, D., He, Y., Xiong, Y., Ye, C., Xu, H.: Neural topic modeling with bidirectional adversarial training. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 340–350 (2020)
29. Wang, X., Yang, Y.: Neural topic model with attention for supervised learning. In: International Conference on Artificial Intelligence and Statistics. pp. 1147–1156. PMLR (2020)
30. Yang, Y., Yao, Q., Qu, H.: Vistopic: A visual analytics system for making sense of large document collections using hierarchical topic modeling. Visual Informatics **1**(1), 40–47 (2017)