Learning Dynamic Hierarchical Topic Graph with Graph Convolutional Network for Document Classification

Zhengjue Wang*[1], Chaojie Wang*[1], Hao Zhang[1], Zhibin Duan[1], Mingyuan Zhou[2], Bo Chen†[1]
[1]National Laboratory of Radar Signal Processing, Xidian University, Xi'an, China
[2]McCombs School of Business The University of Texas at Austin, Austin, TX 78712, USA

Abstract

Constructing a graph with graph convolutional network (GCN) to explore the relational structure of the data has attracted lots of interests in various tasks. However, for document classification, existing graph based methods often focus on the straightforward word-word and word-document relations, ignoring the hierarchical semantics. Besides, the graph construction is often independent from the task-specific GCN learning. To address these constrains, we integrate a probabilistic deep topic model into graph construction, and propose a novel trainable hierarchical topic graph (HTG), including word-level, hierarchical topic-level and document-level nodes, exhibiting semantic variation from fine-grained to coarse. Regarding the document classification as a document-node label generation task, HTG can be dynamically evolved with GCN by performing variational inference, which leads to an end-to-end document classification method, named dynamic HTG (DHTG). Besides achieving state-of-the-art classification results, our model learns an interpretable document graph with meaningful node embeddings and semantic edges.

1 Introduction

Document classification, widely used in numerous downstream applications such as news filtering, spam detection, and recommend system, is a fundamental problem in natural language processing. The basic and essential step for document classification is to extract effective document features. Traditional methods represent documents as sparse lexical vectors, such as bag-of-words (BOW), term frequency-inverse document frequency (TD-IDF), and n-grams. Based on BOW, sophisticated topic models (Blei et al., 2003; Zhou et al., 2015) are developed to explore global semantic characteristics of a document. Considering the sequential information in context, deep learning models including convolutional neural networks (CNNs) (Kim, 2014; Zhang et al., 2015), and recurrent neural networks (RNNs) (Tai et al., 2015; Liu et al., 2016) have been widely used. Though effective, those CNN and RNN models have difficulty in capturing the long-distance relationship. With attention mechanism, Transfomer (Vaswani et al., 2017) is developed to extract global dependencies among all words, whose variants, e.g., BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019), have been used in document classification and achieved promising improvement. However, these deep learning based models pay more attentions on word-level correlations, less considering higher level structures, e.g., semantic-level relations.

Graph is proficient in describing the relations among objects, which has complex topology structure. There is an increasing trend that applying graph convolutional networks (GCNs) on the graph-structured data to further propagate information for various tasks, exhibiting promising performances (Kipf & Welling, 2016; Battaglia et al., 2018; Cai et al., 2018). Although some data are naturally performed in graph structure, e.g., social network, most cases are "non-structural" where the relational structure is not explicit, document data included. Based on raw document data, Yao et al. (2019); Liu et al. (2019); Li et al. (2019) tried to build graphs and fed the graph into a GCN to accomplish text classification, document matching, and article comment generation, respectively. Though achieving appealing results, the process of constructing document graph is usually based on a hand-crafted criterion separated from the GCN learning, which may hurt the performance since the presented graph is not associated with the task closely. Besides, treating words or documents

* Equal Contribution. † Corresponding author.
Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

as nodes, and setting up edges using heuristic distance or words co-occurrence statistics (WCS) in a local window tends to lack semantic consideration. As a result, the constructed document graphs mentioned above often exhibit "flat" structures, hard to model semantic relations at different levels.

Inspired by their works and to beyond these constrains, we integrate the topic model into graph construction, and propose a novel trainable hierarchical topic graph (HTG) cooperated with GCN, leading to an end-to-end document classification method, named dynamic HTG (DHTG). To the best of our knowledge, this is the first effort to combine topic model with GCN for document classification. The following key components constitute our model:

- The proposed HTG contains word-level, hierarchical topic-level, and document-level nodes, representing the semantic variation from fine-grained to coarse.
- Besides traditional static edges, we leverage topic model to construct dynamic ones. Specifically, the advanced topic model in (Zhou et al., 2015) factorizes a BOW vector under Poisson-Gamma distribution as hierarchical products of topic-word representations and document-topic proportions, which are employed to construct the dynamic edges of HTG.
- We regard the document classification as a label generation task w.r.t. the document-level nodes, which is realized by GCN with HTG.
- In order to simultaneously consider the document generative process via the topic model and label generative process through GCN, we maximize the evidence lower bound (ELBO) of joint likelihood via variational inference, realizing dynamically development of HTG and GCN learning.

Besides achieving state-of-the-art classification results, our model learns an interpretable document graph with meaningful node embeddings and semantic edges.

2 Related work

2.1 Sequential neural network for document classification

Considering the sequential relations among local words, two kinds of representative deep neural networks, CNNs and RNNs, are developed for document classification, which mainly based on efficient word embedding. Specifically, Kim (2014) and Kalchbrenner et al. (2014) apply a one-dimension convolutional layer on the word embedding directly, where the filter window acts as a detector of typical n-grams. Beyond them to go deeper, Zhang et al. (2015) and Conneau et al. (2016) design character level CNNs with promising results. In order to construct models with longer memory or scope than CNN, some LSTM based approaches (Tai et al., 2015; Liu et al., 2016) have been presented. In order to describe the relations between words more flexibly, attention mechanisms are introduced into the sequential neural networks for document classification (Yang et al., 2016; Wang et al., 2016). Although these methods are effective in describing local word relationship, all of them ignore the semantic information, e.g., global word occurrence, which is important for document classification, especially for long documents. On the contrary, the proposed HTG not only focuses on the local word dependency through point-wise mutual information (PMI), but also pays more attention on the global hierarchical semantic relations, thanks to the combination with a deep topic model.

2.2 Topic model for document classification

Topic model is proficient to explore the hidden semantic structures of text in an unsupervised manner, which use BOW features as input to extract topics and topic proportion (document feature). Based on a very fundamental topic model, latent Dirichlet allocation (Blei et al., 2003), supervised topic models (Zhu et al., 2012; Lacoste-Julien et al., 2009; Korshunova et al., 2019) are developed in various ways to learn more discriminative features. To alleviate the representation constraint of shallow topic models, Poisson gamma belief network (PGBN) (Zhou et al., 2015) is introduced to build a hierarchical topic model with strong nonlinearity and readily interpretable multilayer latent representations. However, the complex inference procedure of PGBN makes it difficult to perform supervised learning. Derived from PGBN, DHTG reserves the good interpretability of PGBN, but reinvents the extracted information of PGBN to build a structural tree-like document graph with more evident relationships among documents, topics, and words, which is a GCN learning model easier to be supervised inferred.

2.3 Graph representation for document

Constructing a graph with GCN learning was proven to be effective in exploring the relational structure among data in various tasks. There are mainly two ways to organize the raw documents into a graph. In the first type, nodes are word-level, which appear as a key word (Li et al., 2019) or a concept made up by several related words (Liu et al., 2019). The second type not only regards each word as a node, but also represents each document as a node, such that the document classification task can be realized via node classification. TextGCN (Yao et al., 2019), representative in the second type, is most related to this paper. The major differences are: 1) textGCN only cares about relationships among words and documents, ignoring the semantic concepts such as topics; 2) the edges be-

tween nodes in textGCN are predefined static ones, while DHTG builds a dynamic graph developed with GCN learning.

3 Dynamic hierarchical topic graph

In this section, we first introduce the basic background of graph and GCN. Based on it, we present the hierarchical topic graph (HTG), modeling not only document-word and word-word relations like Yao et al. (2019), but also document-topic, topic-topic, and word-topic relations for document classification. By integrating the document generation and classification into a unified framework, the proposed HTG is dynamically evovled with the GCN learning.

3.1 Graph and GCN

Graph is an effective data representation to explore the internal structural relationship existing in the raw data. Formally, a graph can be represented as $G = (V, E)$, where V and E denote the sets of nodes and edges, respectively. In general, each node is assumed to be connected to itself, i.e., $(v, v) \in E$, and the connection between different nodes depends on the data. Denote $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$ as the adjacent matrix of G, whose element $A_{ij} \geq 0$ denotes the connected weight between node i and j. Let $\mathbf{T} = \{\boldsymbol{t}_v\}_{v=1}^{|V|} \in \mathbb{R}^{|V|\times m_0}$ to be a matrix containing all nodes with their original features, where m_0 is the feature size. Having obtained G, GCN (Kipf & Welling, 2016) is an efficient multilayer network to analyze the graph for downstream tasks, via continuous stack of first-order spectral filters followed by a nonlinear activation function.

Specifically, with defined degree matrix $\mathbf{Q}$, where $Q_{ii} = \sum_j A_{ij}$, a basic GCN layer can be represented as

$$\mathbf{H}^{(l)} = \rho(\tilde{\mathbf{A}}\mathbf{H}^{(l-1)}\mathbf{W}_G^{(l)}), l \geq 0 \tag{1}$$

where $\mathbf{H}^{(l)}$ represents the node embedding at layer l with $\mathbf{H}^{(0)} = \mathbf{T}$, $\tilde{\mathbf{A}} = \mathbf{Q}^{-\frac{1}{2}}\mathbf{A}\mathbf{Q}^{-\frac{1}{2}}$ is the normalized symmetric adjacent matrix which is shared in all layers, $\mathbf{W}_G^{(l)} \in \mathbb{R}^{m_{l-1}\times m_l}$ is the GCN filter with m_l denoting the node embedding dimension at layer l, and $\rho(\cdot)$ is the nonlinear activation function such as Relu.

Overall, the construction of the adjacent matrix $\mathbf{A}$ is a crucial problem for GCN, which directly influences the aggregation of information contained in the graph.

3.2 Graph construction

As shown in Fig. 1(a), we build a heterogeneous document graph containing word-level, multiple topic-level and document-level nodes in a hierarchical architecture. Under the assumptions that D denotes the vocabulary size, that K_l denotes the number of topics at layer l, and that N denotes the number of documents, there are $|V| = D + \sum_l K_l + N$ nodes in the document graph, connected by of edges with different semantics. The weights of edges are determined by two kinds of information. We use static information introduced in Yao et al. (2019) to construct the document-word and word-word edges, which are predefined before GCN learning. More importantly, we leverage an probabilistic deep topic model to extract semantic information which is further reinvented to construct word-topic, topic-topic, and document-topic edges. Further, in order to dynamically update the edges to match the classification task better, we combine the topic model and GCN learning under a joint inference framework in a synergistic manner, which improves the performance on the discriminant tasks. Detailed construction process are introduced as follows.

Static edges with predefined weights. According to Yao et al. (2019), TF-IDF indicates the importance of every word in the document, which is proper to define the document-word edges. In order to calculate the weights of word-word edges, they employ PMI with a fixed-size sliding window. Specifically, the PMI of words i and word j is computed as

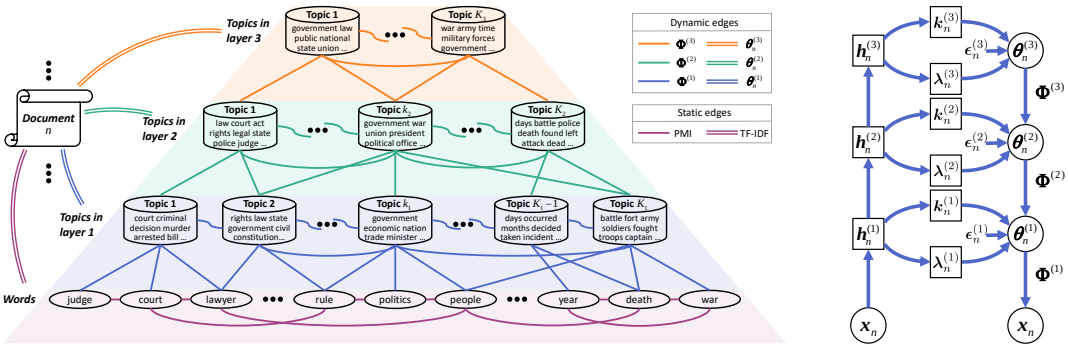
$$\text{PMI}(i, j) = \log \frac{p(i, j)}{p(i)p(j)} \tag{2}$$

$$p(i, j) = \frac{\text{R}(i, j)}{\text{R}}, \quad p(i) = \frac{\text{R}(i)}{\text{R}} \tag{3}$$

where $\text{R}(i, j)$ or $\text{R}(i)$ represents the number of sliding windows in all N documents that contains both word i and j or only word i, respectively, R is the total number of sliding windows. As Yao et al. (2019) discussed, PMI describes the word co-occurrence statistics in a local window. However, they ignore the statistics in a whole document and the global semantic information, which is important for the document classification (Zhu et al., 2012; Mcauliffe & Blei, 2008), especially for the long ones. In the next part, we address it through integrating a deep topic model into graph construction.

Dynamic edges defined by probabilistic deep topic model. In (Zhou et al., 2015), an advanced probabilistic deep topic model, Poisson gamma belief network (PGBN) is proposed to analyze the semantic structure of documents. Consider a corpus $\{\boldsymbol{x}_n\}_{n=1}^N$ containing N documents, where $\boldsymbol{x}_n \in \mathbb{Z}^D$ denotes a BOW count vector. PGBN assumes that $\boldsymbol{x}_n$ is generated from a Poisson likelihood with L Gamma hidden layers, from top to bottom expressed as

$$\begin{aligned}
&\boldsymbol{\theta}_n^{(L)} \sim \text{Gam}\left(\mathbf{1}, 1/c_n^{(L+1)}\right) \\
&\boldsymbol{\theta}_n^{(l)} \sim \text{Gam}\left(\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, 1/c_n^{(l+1)}\right), l = 1, \cdots, L-1 \\
&\boldsymbol{x}_n \sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right),
\end{aligned} \tag{4}$$



(a) (b)

Figure 1: (a) is the architecture of the proposed HTG, where different color backgrounds represent different node layers, corresponding to word-level and hierarchical topic-level from low to high. Moreover, lines with different color or style indicate that the edges in HTG are defined according to different information, which are listed in the legend briefly and discussed in (6) detailedly; (b) is the Weibull upward-downward variational encoder to perform fast inference for document-topic edges $\boldsymbol{\theta}_n^{(l)}$ in (9), where $\boldsymbol{k}_n^{(l)}$ and $\boldsymbol{\lambda}_n^{(l)}$ are deterministically transformed from $\boldsymbol{x}_n$ using a learnable neural network, $\boldsymbol{\epsilon}_n^{(l)}$ is the noise drawn from standard uniform distribution to reparameterize the Weibull variational distribution.

where, $\boldsymbol{\theta}_n^{(l)} \in \mathbb{R}_+^{K_l}$ is the hidden representation in layer l, and $\boldsymbol{\Phi}^{(l+1)} \in \mathbb{R}_+^{K_l \times K_{l+1}}$ is the factor loading matrix in layer l. For scale identifiability and ease of interpretation, PGBN places a simplex constraint on each column of $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$ via a Dirichlet prior, e.g., the k-th column $\boldsymbol{\Phi}_{:k}^{(l)} \sim \mathrm{Dir}(\eta \mathbf{I}_{K_{l-1}})$, where $\mathbf{I}_{K_{l-1}}$ is a vector of K_{l-1} ones.

Under the law of total expectation:

$$\mathbb{E}\left[\boldsymbol{x}_n \,|\, \boldsymbol{\theta}_n^{(l)}, \left\{\boldsymbol{\Phi}^{(t)}, c_n^{(t)}\right\}_{t=1}^{l}\right] = \left[\prod_{t=1}^{l} \boldsymbol{\Phi}^{(t)}\right] \frac{\boldsymbol{\theta}_n^{(l)}}{\prod_{t=2}^{l} c_n^{(t)}}, \quad (5)$$

it is natural to regard $\prod_{t=1}^{l-1} \boldsymbol{\Phi}^{(t)} \boldsymbol{\Phi}_{:k}^{(l)}$ as topic k at layer l, and $\boldsymbol{\theta}_n^{(l)}$ as the topic proportion of document n, respectively. Obviously, the first-layer topics denote the combination of words with weight $\boldsymbol{\Phi}^{(1)}$, while the higher-level topics denote the combination of the lower-level topics with weights $\boldsymbol{\Phi}^{(l)}$. Therefore, these topics describe the global word co-occurrence statistics and hierarchical semantics from detailed to coarse. Inspired by these unique properties, we reinvent these semantic features and integrate PGBN into HTG.

In detail, we construct topic nodes whose semantic definition is the same as the topics of PGBN. Therefore, we use $\boldsymbol{\Phi}^{(1)}$ to define the weighted word-topic edges at layer 1, and use $\{\boldsymbol{\Phi}^{(l)}\}_{l\geq 2}$ to define the weighted topic-topic edges between layer $l-1$ and layer l. Topic proportion $\boldsymbol{\theta}_n^{(l)}$ is used to define edges between document n and topics at layer l. In addition, the cosine distance between $\boldsymbol{\Phi}_{:i}^{(l)}$ and $\boldsymbol{\Phi}_{:j}^{(l)}$ is used to calculated the weight between topics at the same layer. Clearly, these edges are trainable, which can be dynamically evolved during the learning procedure, detailed introduced in Section 3.4.

To sum up, the weighted edges of HTG shown in Fig. 1(a) are defined as:

$$A_{ij} = \begin{cases} \mathrm{PMI}(i,j) & i,j : \mathcal{W} \\ \mathrm{TF\text{-}IDF}(i,j) & i : \mathcal{D}, j : \mathcal{W} \\ \Phi_{ij}^{(1)} & i : \mathcal{W}, j : \mathcal{T} \text{ at layer 1} \\ \Phi_{ij}^{(l)}, l \geq 2 & i,j : \mathcal{T} \text{ at layer } l-1 \text{ and } l \\ \cos(\boldsymbol{\Phi}_{:i}^{(l)}, \boldsymbol{\Phi}_{:j}^{(l)}), l \geq 1 & i,j : \mathcal{T} \text{ at layer } l \\ \theta_{ij}^{(l)}, l \geq 1 & i : \mathcal{D}, j : \mathcal{T} \text{ at layer } l \\ 1 & i = j \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where $\mathcal{W}$, $\mathcal{T}$, and $\mathcal{D}$ represent word, topic and document, respectively.

3.3 Label generation based on GCN

We use GCN to propagate the information in HTG to obtain better node embeddings, where the document-node embeddings are used to perform label generation.

Specifically, assume the corpus has C classes, where $y_n \in \{1, 2, \cdots, C\}$ is the label of document n. Following Yao et al. (2019), we set the original feature matrix $\mathbf{H}^{(0)}$ as identity matrix, i.e., every node is represented as a one-hot vector. And then, we construct a two-layer GCN to obtain the node embedding matrix $\mathbf{H}^{(2)} \in \mathbb{R}^{|V| \times m_2}$, formally stated as

$$\mathbf{H}^{(2)} = \mathrm{GCN}(\mathbf{H}^{(0)}; \{\mathbf{W}_G^{(l)}\}_{l=1}^{2}), \quad (7)$$

where the document-node embeddings are denoted as $\hat{\mathbf{H}}^{(2)} = \{\hat{\boldsymbol{h}}_n^{(2)}\}_{n=1}^{N} \in \mathbb{R}^{N \times m_2}$, with $\hat{\boldsymbol{h}}_n^{(2)}$ corresponding to the n-th document. In this way, $\hat{\mathbf{H}}^{(2)}$ is further mapped for document label generation.

We assume label y_n is generated from a categorical

distribution $y_n \sim \text{Cat}(p_{n1}, \cdots, p_{nC})$, i.e.,

$$p(y_n) = \prod_{c=1}^{C} p_{nc}^{\delta(y_n=c)}, \tag{8}$$

where p_{nc} is the probability that $\hat{\boldsymbol{h}}_n^{(2)}$ belongs to class c, $\delta(\cdot)$ is an indicator function that is equal to one if the argument is true, or zero otherwise. We use a softmax function parameterized by $\mathbf{W}_c \in \mathbb{R}^{m_2 \times C}$ to map $\tilde{\boldsymbol{h}}_n^{(2)}$ to its label probability vector $\boldsymbol{p}_n = (p_{n1}, \cdots, p_{nC})$.

3.4 Joint inference of HTG and GCN

In Yao et al. (2019); Liu et al. (2019); Li et al. (2019), graph construction and GCN learning are separated, which leads to the fact that the task-specified loss of GCN learning can not affect the graph construction. In this section, we propose a variational inference method to infer HTG and GCN jointly. In detail, we need to infer the GCN parameters $\{\mathbf{W}_G^{(l)}\}_{l=1}^2$ and $\mathbf{W}_c$, and the posteriors of $\{\boldsymbol{\Phi}^{(l)}, \boldsymbol{\theta}_n^{(l)}\}_{l=1}^L$ for HTG.

To perform fast inference for document-specific topic proportion $\boldsymbol{\theta}_n^{(l)}$, we employ Weibull upward-downward variational encoder (Zhang et al., 2018) shown in Fig. 1(b) to build the variational posterior of $\boldsymbol{\theta}_n^{(l)}$ as

$$q\left(\boldsymbol{\theta}_n^{(l)}\right) = \text{Weibull}(\boldsymbol{k}_n^{(l)} + \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, \boldsymbol{\lambda}_n^{(l)}; \mathbf{W}_e), \tag{9}$$

where $\boldsymbol{k}_n^{(l)} \in \mathbb{R}^{K_l}$ and $\boldsymbol{\lambda}_n^{(l)} \in \mathbb{R}^{K_l}$ are deterministically transformed from $\boldsymbol{x}_n$ using a neural network parameterized by $\mathbf{W}_e$. We select Weibull variational distribution since 1) it is able to model sparse and non-negative topic proportion, and 2) it is easy to be reparameterized by a noise $\epsilon \sim \text{Uniform}(0, 1)$ (Zhang et al., 2018).

For $\boldsymbol{\Phi}^{(l)}$, we build a Dirichlet variational distribution $q(\boldsymbol{\Phi}^{(l)}) = \text{Dir}(\exp(\boldsymbol{\eta}^{(l)}))$ to ensure the simplex constrain on each column of $\boldsymbol{\Phi}^{(l)}$, where the $\exp(\cdot)$ ensures the values to be positive.

With these variational posteriors and generative processes of documents in (4) and labels in (8), one is able to maximize the ELBO of $\ln p(\boldsymbol{x}_n, y_n)$:

$$\text{ELBO} = \sum_{n=1}^{N} \mathbb{E}\left[\ln p\left(\boldsymbol{x}_n \,|\, \boldsymbol{\Phi}^{(1)}, \boldsymbol{\theta}_n^{(1)}\right) + \ln p\left(y_n \,|\, G\right)\right] - \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{E}\left[\frac{q\left(\boldsymbol{\theta}_n^{(l)}\right)}{p\left(\boldsymbol{\theta}_n^{(l)} | \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)}\right)}\right] + \sum_{l=1}^{L} \mathbb{E}\left[\frac{q(\boldsymbol{\Phi}^{(l)})}{p(\boldsymbol{\Phi}^{(l)})}\right], \tag{10}$$

where $\boldsymbol{\Phi}^{(L+1)} := \mathbf{1}$, $\boldsymbol{\theta}_n^{(L+1)} := \emptyset$, the expectations $\mathbb{E}$ are taken w.r.t. $\{q(\boldsymbol{\theta}_n^{(l)})q(\boldsymbol{\Phi}^{(l)})\}_{l=1}^L$, which are approximated by one sample, G is the proposed HTG.

Even though having achieved the state-of-the-art performance via optimizing (10), we find that the classification accuracy can be improved further if we apply a label regularization on the weighted edge of document-topics, that is topic proportion $\{\boldsymbol{\theta}_n^{(l)}\}$. The objective function is changed as maximizing:

$$\mathcal{L} = \text{ELBO} + \lambda \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{E}_{q(\boldsymbol{\theta}_n^{(l)})}\left[\ln p(y_n \,|\, \boldsymbol{\theta}_n^{(l)})\right], \tag{11}$$

where λ is the penalty parameter which is set as 0.1 in experiments, and the mapping from $\boldsymbol{\theta}_n^{(l)}$ to p_{nc} in (8) is built by another softmax function parameterized by $\mathbf{W}_c'$. As a result, all the parameters in our model are $\boldsymbol{\Theta} = \left\{\mathbf{W}_G, \mathbf{W}_e, \mathbf{W}_c, \mathbf{W}_c', \{\boldsymbol{\eta}^{(l)}\}_{l=1}^L\right\}$. Note that the updates of $\boldsymbol{\Phi}$ and $\boldsymbol{\theta}$ are related not only to the document generative process of topic model, but also to the GCN label generation, which reinvent and extend the hierarchical structure of PGBN into a dynamic document graph for better classification by GCN.

Except for $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^L$, the gradient of $\mathcal{L}$ w.r.t. other parameters are calculated by standard back-propagation. Since the Dirichlet variable is difficult to be reparameterized (Kingma & Welling, 2013), calculating the gradient of $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^L$ is not straightforward. In order to obtain $\nabla_{\boldsymbol{\eta}^{(l)}}\mathcal{L}$ with low variance, we employ General and One-sample (GO) gradient (Cong et al., 2019), as discussed in Appendix A. The whole algorithm is presented in Appendix B.

4 Experiments

Datasets. To evaluate the effectiveness and efficiency of the proposed model, comparison experiments are performed on five widely used document classification datasets, including 20-Newsgroups (20NG), R52, R8, Ohsumed and Movie Review (MR).

- The 20NG dataset[1] contains 18,846 documents from 20 categories, which are separated as 11,314 training samples and 7,532 testing samples.
- R52 and R8[2] are two collected sets from Reuters 21578. R52 has 52 classes including 6,532 training and 2,568 texting documents, while R8 has 8 classes including 5,485 training and 2,189 testing samples.
- The Ohsumed[3] is a bibliographic dataset of medical literature, where each document has one or more labels. Following Yao et al. (2019), we focus on the single-label documents, including 3,357 training and 4,043 test samples.
- MR[4] is a movie review dataset for binary sentiment classification, where each review only contains one

[1] http://qwone.com/~jason/20Newsgroups/
[2] https://www.cs.umb.edu/~smimarog/textmining/datasets/
[3] http://disi.unitn.it/moschitti/corpora.htm
[4] http://www.cs.cornell.edu/people/pabo/movie-review-data/

Table 1: Test classification accuracy on five datasets, where SHTG represents the static HTG which means the HTG is pre-constrcuted before GCN learning, while DHTG represents the dynamic HTG which means the HTG is developed with GCN learning. L_i represents that the model employs i layers of hierarchical topics. All the results of compared methods are provided in Yao et al. (2019). The experiments are run 10 times to get mean ± stand deviation (%).

Model	20NG	R8	R52	Ohsumed	MR
TF-IDF+LR	83.19 ± 0.00	93.74 ± 0.00	86.95 ± 0.00	54.66 ± 0.00	74.59 ± 0.00
PGBN-L_3+LR	83.62 ± 0.21	94.93 ± 0.09	87.36 ± 0.16	56.03 ± 0.45	61.53 ± 0.37
LSTM	75.43 ± 1.72	96.09 ± 0.19	90.48 ± 0.86	51.10 ± 1.50	77.33 ± 0.89
CNN	82.15 ± 0.52	95.71 ± 0.52	87.59 ± 0.48	58.44 ± 1.06	**77.75 ± 0.72**
fastText	79.38 ± 0.30	96.13 ± 0.21	92.81 ± 0.09	57.70 ± 0.49	75.14 ± 0.20
SWEM	85.16 ± 0.29	95.32 ± 0.26	92.94 ± 0.24	63.12 ± 0.55	76.65 ± 0.63
Graph-CNN	81.42 ± 0.32	96.99 ± 0.12	92.75 ± 0.22	63.86 ± 0.53	77.22 ± 0.27
textGCN	86.34 ± 0.09	97.07 ± 0.10	93.56 ± 0.18	68.36 ± 0.56	76.74 ± 0.20
SHTG-L_1	86.38 ± 0.08	97.17 ± 0.09	93.58 ± 0.16	67.95 ± 0.46	76.92 ± 0.18
SHTG-L_2	86.43 ± 0.09	97.21 ± 0.08	93.69 ± 0.12	68.38 ± 0.40	76.99 ± 0.16
SHTG-L_3	86.82 ± 0.08	97.24 ± 0.08	93.77 ± 0.10	68.52 ± 0.35	77.04 ± 0.13
DHTG w/o regularization	86.91 ± 0.10	97.29 ± 0.08	93.85 ± 0.11	68.71 ± 0.33	77.10 ± 0.15
DHTG	**87.13 ± 0.07**	**97.33 ± 0.06**	**93.93 ± 0.10**	**68.80 ± 0.33**	77.21 ± 0.11

sentence (Pang & Lee, 2005). We follows the training/testing split in Tang et al. (2015)

Preprocessings. We preprocess all the datasets by cleaning and tokenizing document as Kim (2014); Yao et al. (2019). And then we remove the stop words and low frequency words appearing less than 5 times for all datasets except MR, since the documents in MR are relatively short. The statistics of every dataset after preproceeing are summarized in Appendix C.

Comparison approaches. We compare DHTG with some related document classification models as follows:

- **TF-IDF+LR:** The Logistic Regression is applied on the TF-IDF features of the documents.
- **LSTM (Liu et al., 2016):** The LSTM model with pre-trained word embeddings, where the last hidden states as regarded as features.
- **CNN (Kim, 2014):** A convolutional neural network with pre-trained word embeddings.
- **fastText (Joulin et al., 2016):** The average word embeddings as treated as document embedding, which is fed into a linear classifier.
- **SWEM (Shen et al., 2018):** A word embedding model that employs pooling strategies operated over word embeddings.
- **Graph-CNN (Defferrard et al., 2016):** A graph CNN model that operates convolutions over a word embedding similarity graph.
- **textGCN (Yao et al., 2019):** A GCN model that builds a static graph only with document and word nodes, which can be seen as a part of HTG.
- **PGBN-L_3+LR (Zhou et al., 2015):** The unsupervised 3-layer deep topic models, whose topic proportions of different layers are concatenated to apply Logistic Regression.

Variants. In order to clearly illustrate the motivations and effectiveness of our model, we perform several variants. We use a well-trained PGBN model to build a HTG with i layers of topics, whose construction is static and independent from the GCN learning, termed as SHTG-L_i. We also perform a variant of DHTG without label regularization on $\boldsymbol{\theta}_n^{(l)}$ as Eq. 11, termed as "DHTG w/o regularization".

Settings. Having observed the superior performance of SHTG with 3-layer topics, we set the topic layers of DHTG as 3, where the topic numbers at different layers from low to high are set as 256, 128, and 64, respectively. The PMI window size is set as 20, following Yao et al. (2019). For GCN, the node embedding sizes $\{m_l\}_{l=1}^2$ in (1) are set as 200. We use Adam (Kingma & Ba, 2014) with learning rate 0.02 and dropout rate 0.5 to avoid overfitting. We random select 10% of training data as the validation set to perform model selection.

4.1 Document Classification

Classification accuracy results of all the comparison models on every dataset are concluded in Table 1, which illustrates the outperformance of DHTG with high accuracy and low variance. Among these methods, both TF-IDF+LR and PGBN-L_3+LR apply Logistic Regression to achieve classification, but regard TF-IDF and topic proportion as features, respectively. According to their comparison results, except for MR where the text is too short to learn global semantics, PGBN-L_3+LR performs better than TF-IDF+LR on other datasets, which demonstrates the semantic information expressed by hierarchical topics is more effective for long document classification. On the contrary, LSTM and CNN, capturing sequential information, perform the best on MR, which is likely due to the fact that local word sequence is important in short document. Even so, DHTG assembles many kinds of information

into a complex graph, achieving comparable results on MR. FastText, SWEM, and Graph-CNN concentrate the information of pre-trained word embeddings via average, pooling methods, and building word similarity graph, respectively, showing competitive performances. Beyond them, textGCN takes advantage of both document-word and word-word relations by TF-IDF and PMI to learn word embeddings and integrate them to obtain document embeddings. Besides these relations, DHTG leverages probabilistic deep topic model to obtain semantic information, which is further used to construct word-topic, topic-topic, and document-topic relations. Moreover, different from Graph-CNN and textGCN where the graph construction is independent from the subsequent learning process, DHTG dynamically optimizes the graph with GCN learning.

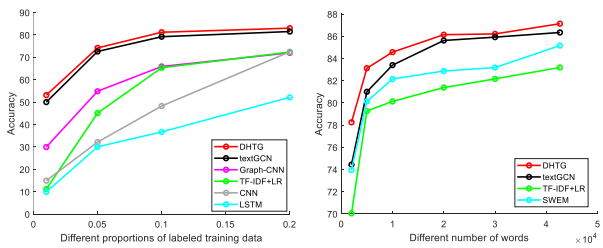
Discussion on variants. According to the results of different SHTG variants shown in Tabel 1, the classification accuracy is improved as the topic structure becomes deeper, which illustrates the effectiveness of hierarchical topic nodes. Similar results on DHTG can be found in Appendix E. Having the same topic layers, DHTG shows higher accuracy than SHTG-L_3, which indicates that the joint learning of HTG and GCN makes the graph match the classification task better. In addition, DHTG performs better than DHTG w/o regularization, which validates the effectiveness of label regularization.

Discussion on effects of the size of labeled data. Following Yao et al. (2019), we test several best performing models with 1%, 5%, 10%, and 20% training data on 20NG, whose results are shown in Fig. 2(a). It can be seen that, with the decrease of the amount of labeled training data, the accuracy results of GCN based models, textGCN and DHTG, drop more slowly than that of others, which illustrates that GCN is able to propagate document label information to the entire graph well as discussed in Kipf & Welling (2016).

Discussion on effects of the size of vocabulary. We also perform the experiments on 20NG with different vocabulary sizes, where we select top $2K$, $5K$, $10K$, $20K$ and $30K$ words according to the frequency. As shown in Fig. 2(b), with limited vocabulary, the accuracy of DHTG drops more slowly than others, which illustrates that, to some extent, the hierarchical topical semantics in DHTG compensate for the information loss brought by deceasing the number of words.

4.2 Visulization of HTG

Semantics among topic nodes. For better understanding the hierarchical semantic relations learned by DHTG, we first focus on the topic nodes and visualize a subgraph on 20NG, as shown in Fig. 4. Clearly, the semantic meaning of each topic-node and the edges



(a) (b)

Figure 2: Test accuracy by different (a) sizes of training data and (b) sizes of vocabulary.

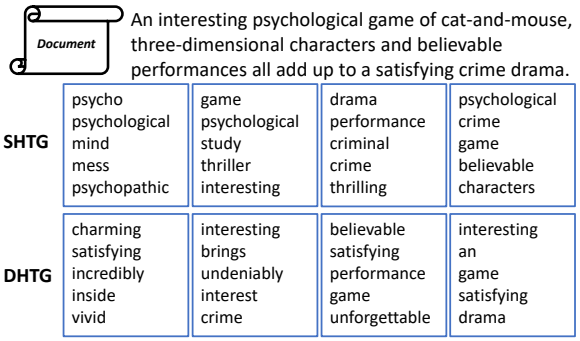


Figure 3: Illustration of the top-4 topics at layer 1 with large document-topic weights, learned by SHTG and DHTG, respectively. This document is a positive review from MR dataset, whose description is shown on the top.

between topics are highly interpretable. Specifically, with the increasing of topic layers, the topic semantics vary from detailed to coarse, due to the fact that higher topics are composed of the lower ones as introduced in (5). Besides, topics at the same layer with similar semantics build edges with higher weights. Moreover, it is worth noting that some groups of topics have no direct connection at the first layer but their combinations at a higher layer build connections, since the higher-layer topics owns more general semantics.

Semantics among document-topic nodes. Focusing on document-topic relations, we randomly select a positive movie review from MR testing dataset and list the top-4 topics at layer 1 with large document-topic weights, learned by SHTG and DHTG, respectively, as shown in Fig. 3. It can be seen that, this review contains both content description, e.g., "psychological game" and "crime drama", and sentiment comments like "interesting" and "satisfying". According to the results, SHTG is able to capture both content and sentiment topics. Influenced by not only document generation but also the label generation as described in (11), the document node of DHTG is impelled to be more related to those sentiment topics rather than content topics. In other words, the document embedding in DHTG is more likely to fuse these sentiment topic embeddings by GCN for better classification. The rela-

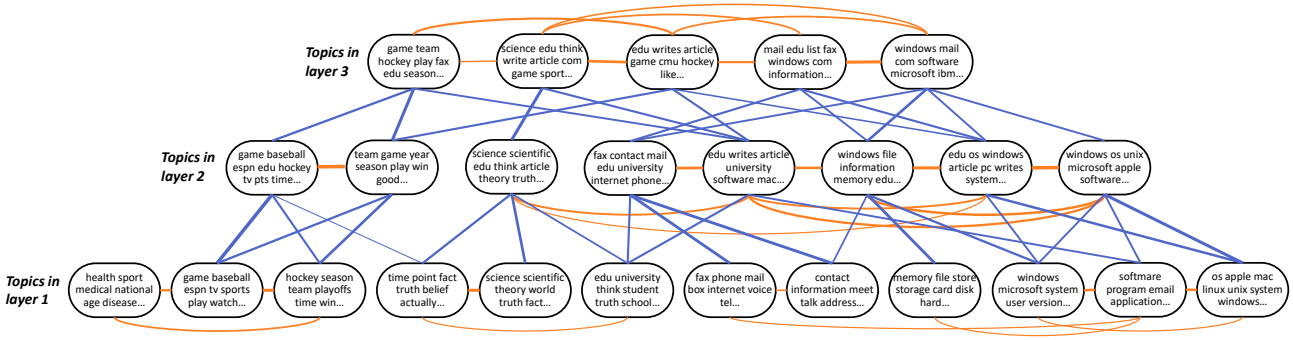


Figure 4: Illustration of the semantic relations among topic nodes, learned by DHTG from 20NG, where each topic is displayed by its most representative words. The thickness of the connecting line corresponds to the weight of the edge.

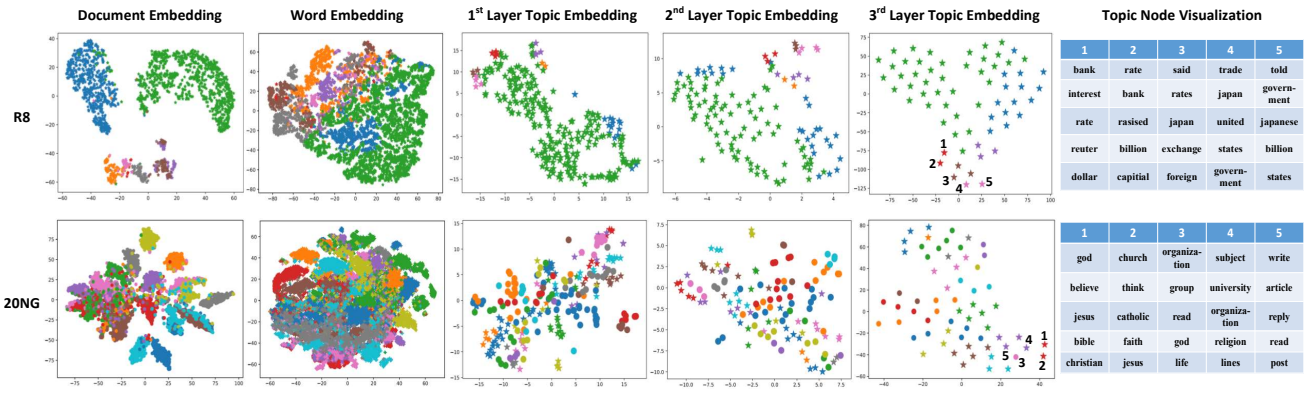


Figure 5: The t-SNE visualization of node embeddings of douments, words, first-layer topics, second-layer topics and third-layer topics from left to right, respectively, where the first-row results are on R8 and the second-row ones are on 20NG. Points with different colors and styles correspond to different classes. The sixth column lists some topics represented by the top-5 words, whose index corresponds to the same one in the fifth column.

tion between document and word nodes also exhibits similar semantic results, which is shown in Appendix D.

Node embedding via t-SNE. We qualitatively visualize different types of node embeddings learned by DHTG, i.e., the output of the second layer of GCN. With the help of t-SNE (Maaten & Hinton, 2008), the visualization results is shown in Fig 5, where the first and second rows correspond to R8 and 20NG, respectively. According to the distribution of document embeddings illustrated by the true label, DHTG learns discriminative document features with good separability. Since the word or topic embeddings are connected with the document embedding by the edges of graph, following (Yao et al., 2019), we regard the word and topic embeddings as a document embedding and use the learned softmax classifier $\mathbf{W}_c$ in (8) to determine their labels. According to the second to fifth columns in Fig 5, it can be seen that word or topic embeddings with the same label are close to each other. A more intuitive illustration is shown by the fifth and sixth column, where the semantic similarity is accord with the distance between node embeddings. On the other hand, word and topic embeddings on different layers exhibit different separability, which are gathered and transmitted to the document embeddings by the graph to promote the classification performance. As shown in the upper part of Fig. 5, an interesting phenomenon is that the imbalance degree of data are alleviated in different semantic layers, which might due to the fact that topic nodes with similar semantics tend to be aggregate in higher semantic layer to exhibit more general topic nodes.

5 Conclusion

With the help of a deep probabilistic topic model, we propose the hierarchical topic graph for document classification containing different types of edges among word-nodes, hierarchical topic-nodes and document-nodes. In order to realize dynamic evolution of HTG, we integrate variational inference and GCN learning into a unified framework, obtaining a model named dynamic HTG (DHTG). Besides achieving state-of-the-art document classification performance even with limited labeled data, our model learns an interpretable document graph with meaningful node embeddings and semantic edges.

6 Acknowledgments

Bo Chen acknowledges the support of the Young Thousand Talent by Chinese Central Government, the 111 Project (No. B18039), NSFC (61771361), NSFC for Distinguished Young Scholars (61525105) and Shaanxi Inovation Team Project. M. Zhou acknowledges the support of Award IIS-1812699 from the U.S. National Science Foundation, and support from the McCombs Research Excellence Grant.

References

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

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

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *CVPR*, pp. 5177–5186, 2019.

Yulai Cong, Miaoyun Zhao, Ke Bai, and Lawrence Carin. Go gradient for expectation-based objectives. In *ICLR*, 2019.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pp. 3844–3852, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Keith O. Geddes, M. Lawrence Glasser, Reg A. Moore, and Tony C. Scott. Evaluation of classes of definite integrals involving elementary functions via differentiation of special functions. *Applicable Algebra in Engineering, Communication and Computing*, 1(2): 149–165, 1990.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *EACL*, 2016.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

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

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

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Iryna Korshunova, Hanchen Xiong, Mateusz Fedoryszak, and Lucas Theis. Discriminative topic modeling with logistic lda. In *NIPS*, 2019.

Simon Lacoste-Julien, Fei Sha, and Michael I Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In *NIPS*, pp. 897–904, 2009.

Wei Li, Jingjing Xu, Yancheng He, Shengli Yan, Yunfang Wu, et al. Coherent comment generation for chinese articles with a graph-to-sequence model. *arXiv preprint arXiv:1906.01231*, 2019.

Bang Liu, Di Niu, Haojie Wei, Jinghong Lin, Yancheng He, Kunfeng Lai, and Yu Xu. Matching article pairs with graphical decomposition and convolutions. In *ACL*, pp. 6284–6294, 2019.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Jon D Mcauliffe and David M Blei. Supervised topic models. In *NIPS*, pp. 121–128, 2008.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pp. 115–124, 2005.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *ACL*, pp. 440–450, 2018.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pp. 5998–6008, 2017.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*, pp. 606–615, 2016.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *NAACL*, pp. 1480–1489, 2016.

Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *AAAI*, volume 33, pp. 7370–7377, 2019.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, pp. 649–657, 2015.

Mingyuan Zhou, Yulai Cong, and Bo Chen. The poisson gamma belief network. In *NIPS*, pp. 3043–3051, 2015.

Jun Zhu, Amr Ahmed, and Eric P Xing. Medlda: maximum margin supervised topic models. *Journal of Machine Learning Research*, 13(Aug):2237–2278, 2012.

A Appendix: GO gradient for $\left\{\nabla_{\boldsymbol{\eta}^{(l)}}\mathcal{L}\right\}_{l=1}^{L}$

Except for the parameters $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^{L}$ of variational distribution $q(\boldsymbol{\Phi}^{(l)}) \sim \mathrm{Dir}(\exp(\boldsymbol{\eta}^{(l)})), l = 1, \cdots, L$, the gradients of other parameters w.r.t. $\mathcal{L}$ in (11) can be easily calculated by standard BP algorithm. For the gradient of $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^{L}$ w.r.t. $\mathcal{L}$, standard BP chain can not be applied straightforward, since Dirichlet distribution can not be reparameterized easily. In order to calculate gradient of $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^{L}$ w.r.t. $\mathcal{L}$ with low variance, we apply GO gradient algorithm (Cong et al., 2019) here, which will be discussed in this Appendix.

With the following meanfield assumption:

$$q\left(\left\{\boldsymbol{\Phi}^{(l)}\right\}_{l=1}^{L}\right) = \prod_{l=1}^{L}\prod_{k=1}^{K_l} q_{\boldsymbol{\eta}_{:k}^{(l)}}\left(\boldsymbol{\Phi}_{:k}^{(l)}\right), \tag{12}$$

where K_l denotes the total numbers of topics at layer l of PGBN, $\boldsymbol{\Phi}_{:k}^{(l)}$ denotes the k-th column of $\boldsymbol{\Phi}^{(l)}$, and $\boldsymbol{\eta}_{:k}^{(l)}$ denotes the k-th column of $\boldsymbol{\eta}^{(l)}$. Therefore, we only need to discuss how to use GO to calculate the gradient $\nabla_{\boldsymbol{\eta}_{:k}^{(l)}}\mathbb{E}_{q_{\boldsymbol{\eta}_{:k}^{(l)}}(\boldsymbol{\Phi}_{:k}^{(l)})}\left[g(\boldsymbol{\Phi}_{:k}^{(l)})\right]$, where g is a function that gets rid of the expectation in (11). For simplicity, we use $\boldsymbol{\eta} \sim \mathbb{R}^{O\times 1}$ to denote $\boldsymbol{\eta}_{:k}^{(l)}$, and $\boldsymbol{\phi} \sim \mathbb{R}^{O\times 1}$ to denote $\boldsymbol{\Phi}_{:k}^{(l)}$. With these illustrations, our core problem is defined as how to calculate

$$\nabla_{\boldsymbol{\eta}}\mathbb{E}_{q_{\boldsymbol{\eta}}(\boldsymbol{\phi})}\left[g(\boldsymbol{\phi})\right] \tag{13}$$

As we know, if $q(\boldsymbol{\phi}) \sim \mathrm{Dir}(\exp(\boldsymbol{\eta}))$, we can sample it as

$$\begin{aligned}&\boldsymbol{\psi}_o \sim \mathrm{Gam}(\exp(\boldsymbol{\eta}_o), 1), o = 1, \cdots, O\\ &\boldsymbol{\phi} = \left[\frac{\boldsymbol{\psi}_1}{\sum_{o=1}^{O}\boldsymbol{\psi}_o}; \cdots; \frac{\boldsymbol{\psi}_O}{\sum_{o=1}^{O}\boldsymbol{\psi}_o}\right].\end{aligned} \tag{14}$$

As a result, the core problem in (13) is changed to

$$\nabla_{\boldsymbol{\eta}}\mathbb{E}_{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}\left[g(\boldsymbol{\psi})\right]. \tag{15}$$

As given in Theorem 1 in Cong et al. (2019), the above gradient can we written as

$$\nabla_{\boldsymbol{\eta}}\mathbb{E}_{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}\left[g(\boldsymbol{\psi})\right] = \mathbb{E}_{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}\left[\mathbb{G}_{\boldsymbol{\eta}}^{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}\mathbb{D}_{\boldsymbol{\psi}}\left[g(\boldsymbol{\psi})\right]\right], \tag{16}$$

where

$$\mathbb{D}_{\boldsymbol{\psi}}\left[g(\boldsymbol{\psi})\right] = \left[\mathbb{D}_{\boldsymbol{\psi}_1}[g(\boldsymbol{\psi})], \cdots, \mathbb{D}_{\boldsymbol{\psi}_o}[g(\boldsymbol{\psi})], \cdots, \mathbb{D}_{\boldsymbol{\psi}_O}[g(\boldsymbol{\psi})]\right]^T$$

with

$$\mathbb{D}_{\boldsymbol{\psi}_o}[g(\boldsymbol{\psi})] \overset{def}{=} \nabla_{\boldsymbol{\psi}_o}g(\boldsymbol{\psi}) \tag{17}$$

which is easy to calculate by stand BP algorithm;

$$\mathbb{G}_{\boldsymbol{\eta}}^{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})} \overset{def}{=} \left[s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_1)}, \cdots, s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_o)}, \cdots, s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_O)}\right], o = 1, \cdots, O,$$

with

$$\begin{aligned}s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_o)} &= \frac{-1}{q(\boldsymbol{\psi}_o)}\nabla_{\boldsymbol{\eta}}Q(\boldsymbol{\psi}_o)\\ &= \frac{-1}{q(\boldsymbol{\psi}_o)}\left[\nabla_{\boldsymbol{\eta}_1}Q(\boldsymbol{\psi}_o), \cdots, \nabla_{\boldsymbol{\eta}_O}Q(\boldsymbol{\psi}_o)\right] \in \mathbb{R}^{O\times 1},\end{aligned} \tag{18}$$

where $Q(\boldsymbol{\psi}_o)$ is the CDF of $q(\boldsymbol{\psi}_o)$. As discussed in (14), $q(\boldsymbol{\psi}_o) \sim \mathrm{Gam}(\exp(\boldsymbol{\eta}_o), 1)$, and each $\{\boldsymbol{\psi}_o\}_{o=1}^{O}$ are independently sampled. Thus, the elements in $s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_o)}$ are all zeros except the o-th element being $s_{\boldsymbol{\eta}_o}^{q(\boldsymbol{\psi}_o)} = \frac{-1}{q(\boldsymbol{\psi}_o)}\nabla_{\boldsymbol{\eta}_o}Q(\boldsymbol{\psi}_o)$. Thus, matrix $\mathbb{G}_{\boldsymbol{\eta}}^{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}$ is a diagonal matrix.

In Cong et al. (2019), if $q(\boldsymbol{\psi}_o)$ is the gamma distribution $\mathrm{Gam}(\alpha, 1)$ where $\alpha = \exp(\boldsymbol{\eta}_o)$, the authors give the result of $s_{\boldsymbol{\eta}_o}^{q(\boldsymbol{\psi}_o)}$ as

$$s_{\boldsymbol{\eta}_o}^{q(\boldsymbol{\psi}_o)} = \frac{[\log(\boldsymbol{\psi}_0) - \digamma(\alpha)]\Gamma(\alpha, \boldsymbol{\psi}_0) + \boldsymbol{\psi}_0\mathcal{T}(3, \alpha, \boldsymbol{\psi}_0)}{\boldsymbol{\psi}_0^{\alpha-1}\exp^{-\boldsymbol{\psi}_0}}, \tag{19}$$

where $\digamma(\cdot)$ is the digamma function, $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function, and $\mathcal{T}(\cdot,\cdot,\cdot)$ is a special case of Meijer G-function (Geddes et al., 1990).

For clearer to understand know how to calculate the gradient of $\left\{\nabla_{\boldsymbol{\eta}^{(l)}}\mathcal{L}\right\}_{l=1}^{L}$, as shown in Fig. 6, we give the illustration of the feed-forward and back-propagation process of parameters $\boldsymbol{\eta}$.

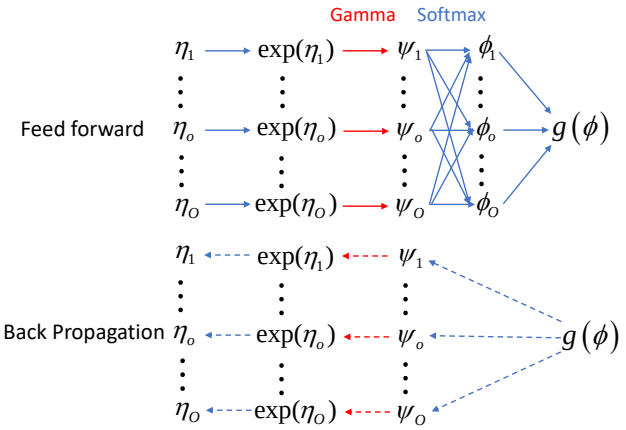


Figure 6: Feed forward and back propagation process of parameters $\boldsymbol{\eta}$, where for feed forward, the blue lines denote deterministic mapping and the red ones denote stochastic sample; for back propagation, the blue lines denote standard BP chain calculated by (17) and the red ones denote the BP calculated by (19).

Table 2: The statstics of each dataset after preprocessing.

Dataset	# Docs	# Training	# Test	# Word	# Nodes	# Classes	Average Length
20NG	18,846	11,314	7,532	42,757	62,051	20	221.26
R8	7,674	5,485	2,189	7,688	15,810	8	65.72
R52	9,100	6,532	2,568	8,892	18,440	52	69.82
Ohsumed	7,400	3,357	4,043	14,157	22,005	23	135.82
MR	10,662	7,108	3,554	18,764	29,874	2	20.39

B Appendix: The whole algorithm of our model

In this paper, different from many existing models (Yao et al., 2019; Liu et al., 2019; Chen et al., 2019; Li et al., 2019) that separate the graph construction and GCN learning, we propose a joint learning method to build graph dynamically with GCN learning based a unified loss function in (11). In Algorithm 1, we give a detailed step to illustrate how to update the models.

Algorithm 1 Joint learning of HTG and GCN.

Initialize WUDVE encoder parameters $\mathbf{W}_e$, GCN parameters $\{\mathbf{W}_G^{(l)}\}_{l=1}^2$, softmax function parameters $\mathbf{W}_c$ and $\mathbf{W}_c^{'}$, variational distribution parameters of topics $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^L$.

for $iter = 1, 2, \cdots$ **do**

1) Randomly select a mini-batch containing $\tilde{N}$ documents with its labels $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^{\tilde{N}}$;

2) Draw random noise $\left\{\boldsymbol{\varepsilon}_n^{(l)}\right\}_{n=1,l=1}^{N,L}$ from uniform distribution to reparameterize the Weibull variational distribution of $\boldsymbol{\theta}_n$;

3) Approximate the expectation in $\mathcal{L}$ (11) by one sample;

4) Calculate $\nabla_{\mathbf{W}_e}\mathcal{L}$, $\nabla_{\mathbf{W}_G}\mathcal{L}$, $\nabla_{\mathbf{W}_c}\mathcal{L}$, $\nabla_{\mathbf{W}_c'}\mathcal{L}$ by standard Back Propagation (BP);

5) Calculate $\left\{\nabla_{\boldsymbol{\eta}^{(l)}}\mathcal{L}\right\}_{l=1}^{\mathcal{L}}$ according to GO gradient (Cong et al., 2019), which is specified in Appendix A.

6) Update $\mathbf{W}_e$,$\{\mathbf{W}_G^{(l)}\}_{l=1}^2$, $\mathbf{W}_c$, $\mathbf{W}_c^{'}$ and $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^L$ through gradient descend algorithm such as ADAM to minimize the loss $-\mathcal{L}$ in (11)).

end for

C Detailed statistics of each dataset

For better understand the statistics of each dataset, the statistics of each dataset after preprocessing are summarized in Table 2. The number of nodes $|V| = D + \sum_{l=1}^3 K_l + N$, where D represents the number of words, K_l represents the number of topics at layer l (we use $K_1 = 256$, $K_2 = 128$, $K_3 = 64$ in all experiments), and N represents the number of training documents, respectively. Note that, in practice at each iteration in Algorithm 1, we only select a mini-batch containing $\tilde{N}$ documents with its labels. This operation can not break our developed HTG due to the fact that one document node only has edges with other type nodes but has no edge with other document nodes.

D Semantics among document-topic nodes of 20NG

We send each word embedding at GCN-layer-2 to the classifier $\mathbf{W}_c$ in (8). In Fig. 7, we show the top 10 words with highest values corresponding to some classes on 20NG. Clearly, we note that the top 10 words are interpretable, which are very close to the label's meaning.

E DHTG with different layers of topics

We use a well-trained GBN model to build a static HTG with different layers, represented as SHTG-L1, SHTG-L2, SHTG-L3, respectively. The comparison results are listed in Table 1. As consistently observed across all datasets, the classification accuracy increases with more layers, illustrating the effectiveness of multi-layer document representations. As a complementary experiment, we build a DHTG with different topic-layers, with the test accuracy results on five datasets listed in Table 3. A similar phenomenon is observed that the classification accuracy increases with more topic layers.

F Another Statistics of test accuracy compare between DHTG and textGCN

In Yao et al. (2019), the mean and standard deviation of the test accuracy is achieved by running 10-times experiments with different weight initialization, but the same training/test split. To make fair comparison with the results listed in Yao et al. (2019), we follow their setting and summarize the results as Table 1. A reviewer suggested us to show the mean and standard

rec.sport.baseball	sci.med	rec.autos	comp.windows.x	soc.religion.christian	talk.politics.guns
hitter	candida	car	windows	church	gun
pitching	disease	cars	dos	jesus	firearms
baseball	patients	v12	exe	christians	ax
braves	vitamin	callison	file	faith	fbi
pitcher	syndrome	engine	win3	bible	handheld
batting	infection	toyota	drivers	christianity	weapons
cubs	chronic	nissan	fonts	catholic	handgun
players	doctor	mustang	files	christian	firearm
phillies	clinical	wagon	font	heaven	amendment
pitchers	hiv	ford	zip	romans	handguns

Figure 7: Words with highest values for several classes in 20NG .

Table 3: Test classification accuracy on five datasets with different layers of PGBN in DHTG.

Model	20NG	R8	R52	Ohsumed	MR
DHTG-L1	86.69 ± 0.08	97.21 ± 0.07	93.67 ± 0.16	68.15 ± 0.40	77.02 ± 0.16
DHTG-L2	86.93 ± 0.07	97.29 ± 0.05	93.81 ± 0.13	68.51 ± 0.36	77.11 ± 0.15
DHTG-L3	87.13 ± 0.07	97.33 ± 0.06	93.93 ± 0.10	68.80 ± 0.33	77.21 ± 0.11

Table 4: Test accuracy of textGCN and DHTG on five dataset, where the mean and the standard deviation are achieved by running 10 times experiments with different weight initialization and different training/test split.

Model	20NG	R8	R52	Ohsumed	MR
textGCN	85.27± 0.36	96.51 ± 0.35	94.07 ± 0.32	68.56 ± 0.52	75.61 ± 0.31
DHTG	86.85 ± 0.23	97.10 ± 0.19	94.33 ± 0.28	69.08 ± 0.41	77.10 ± 0.19

deviation of the test accuracy by running 10-times experiments with different weight initialization and different training/test split. In Table 4, we give the corresponding results. Clearly, compared with textGCN, DHTG has higher mean and lower standard deviation, demonstrating the superior performance of DHTG.