Inductive Topic Variational Graph Auto-Encoder for Text Classification

Qianqian Xie
Department of Computer Science,
University of Manchester
qianqian.xie@manchester.ac.uk

Jimin Huang
School of Computer Science,
Wuhan University
huangjimin@whu.edu.cn

Pan Du
Department of Computer Science and Operations Research, University of Montreal
pandu@iro.umontreal.ca

Min Peng
School of Computer Science,
Wuhan University
pengm@whu.edu.cn

Jian-Yun Nie
Department of Computer Science and Operations Research,
University of Montreal
nie@iro.umontreal.ca

Abstract

Graph convolutional networks (GCNs) have been applied recently to text classification and produced an excellent performance. However, existing GCN-based methods do not assume an explicit latent semantic structure of documents, making learned representations less effective and difficult to interpret. They are also transductive in nature, thus cannot handle out-of-graph documents. To address these issues, we propose a novel model named inductive Topic Variational Graph Auto-Encoder (T-VGAE), which incorporates a topic model into variational graph-auto-encoder (VGAE) to capture the hidden semantic information between documents and words. T-VGAE inherits the interpretability of the topic model and the efficient information propagation mechanism of VGAE. It learns probabilistic representations of words and documents by jointly encoding and reconstructing the global word-level graph and bipartite graphs of documents, where each document is considered individually and decoupled from the global correlation graph so as to enable inductive learning. Our experiments on several benchmark datasets show that our method outperforms the existing competitive models on supervised and semi-supervised text classification, as well as unsupervised text representation learning. In addition, it has higher interpretability and is able to deal with unseen documents.

1 Introduction

Recently, graph convolutional networks (GCNs)(Kipf and Welling, 2017; Veličković et al., 2018) have been successfully applied to text classification tasks (Peng et al., 2018a; Yao et al., 2019; Liu et al., 2020; Wang et al., 2020). In addition to the local information captured by CNN or RNN, GCNs learn word and document representations by taking into account the global correlation information embedded in the corpus-level graph, where words and documents are nodes connected by indexing or citation relations.

However, the **hidden semantic structures**, such as latent topics in documents (Blei et al., 2003; Yan et al., 2013; Peng et al., 2018b), is still ignored by most of these methods (Yao et al., 2019; Huang et al., 2019; Liu et al., 2020; Zhang et al., 2020), which can improve the text representation and provide extra **interpretability** (in which the probabilistic generative process and topics make more sense to humans compared to neural networks, i.e. topics can be visually represented by top-10 or 20 most probable word clusters). Although few studies such as (Wang et al., 2020) have proposed incorporating a topic structure into GCNs, the topics are extracted in advance from the set of documents, independently from the graph and information propagation among documents and words. We believe that the topics should be determined in accordance with the connections in the graph. For example, the fact that two words are connected provides extra information that these words are on a similar topic(s). Moreover, existing GCN-based methods are limited by their transductive learning nature, i.e. a document can be classified only if it is already seen in the training phase (Wang et al., 2020; Yao et al., 2019; Liu et al., 2020). The lack of **inductive learning** ability for unseen documents is a critical issue in practical text classification applications, where we have to deal with new documents. It is

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4218–4227
June 6–11, 2021.

Table 1: Comparison with related work. We compare the manner of model learning, whether incorporate the latent topic structure and the manner of topic learning of these models.

Model	Explainability	Learning	Topics
TextGCN (Yao et al., 2019)	-	transductive	-
TensorGCN (Liu et al., 2020)	-	transductive	-
DHTG (Wang et al., 2020)	√	transductive	static
T-GCN (Huang et al., 2019)	-	inductive	-
TG-Trans (Zhang and Zhang, 2020)	-	inductive	-
TextING (Zhang et al., 2020)	-	inductive	-
HyperGAT (Ding et al., 2020)	-	inductive	-
Our model	√	inductive	dynamic

intuitive to decouple documents with the global graph and treat each document as an independent graph (Huang et al., 2019; Zhang et al., 2020; Ding et al., 2020; Zhang and Zhang, 2020; Xie et al., 2021). However, no attempt has been made to address both aforementioned issues.

To address these issues, we incorporate the topic model into variational graph auto-encoder (VGAE), and propose a novel framework named inductive Topic Variational Graph Auto-Encoder (T-VGAE). T-VGAE first learns to represent the words in a latent topic space by embedding and reconstructing the word correlation graph with the GCN probabilistic encoder and probabilistic decoder. Take the learned word representations as input, a GCN-based message passing probabilistic encoder is adopted to generate document representations via information propagation between words and documents in the bipartite graph. We compare our model with existing related work in Table 1. Different from previous approaches, our method unifies topic mining and graph embedding learning with VGAE, thus can fully embed the relations between documents and words into dynamic topics and provide interpretable topic structures into representations. Besides, our model builds a document-independent word correlation graph and a word-document bipartite graph for each document instead of a corpus-level graph to enable inductive learning.

The main contributions of our work are as follows:

1. We propose a novel model T-VGAE based on topic models and VGAE, which incorporates latent topic structures for inductively document and word representation learning. This makes the model more effective and interpretable.

2. we propose to utilize the auto-encoding variational Bayes (AEVB) method to make efficient black-box inference of our model.

3. Experimental results on benchmark datasets demonstrate that our method outperforms the existing competitive GCN-based methods on supervised and semi-supervised text classification tasks. It also outperforms topic models on unsupervised text representation learning.

2 Related Work

2.1 Graph based Text Classification

Recently, GCNs have been applied to various NLP tasks (Zhang et al., 2018; Vashishth et al., 2019). For example, TextGCN (Yao et al., 2019) was proposed for text classification, which enriches the corpus-level graph with the global semantic information to learn word and document embeddings. Inspired by it, Liu et al. (Liu et al., 2020) further considered syntactic and sequential contextual information and proposed TensorGCN. However, none of them utilized the latent semantic structures in the documents to enhance text classification. To address the issue, (Wang et al., 2020) proposed dynamic HTG (DHTG), in an attempt to integrate the topic model into graph construction. DHTG learned latent topics from the document-word correlation information (similar to traditional topic models), which will be used for GCN based document embedding. However, the topics in DHTG were learned independently from the word relation graph and the information propagation process in the graph, in which word relations are ignored.

Moreover, the existing GCN-based methods also require a pre-defined graph with all the documents and cannot handle out-of-graph documents, thus limiting their practical applicability.

To deal with the inductive learning problem, (Huang et al., 2019; Zhang et al., 2020; Ding et al., 2020; Zhang and Zhang, 2020) proposed to consider each document as an independent graph for text classification. However, the latent semantic structure and interpretability are still ignored in these methods. Different from previous approaches, we aim to deal with both issues of dynamic topic structure and inductive learning. We propose to combine the topic model and graph based information propagation in a unified framework with VGAE to learn interpretable representations for words and documents.

2.2 Graph Enhanced Topic Models

There are also studies trying to enhance topic models with efficient message passing in the graph data structure of GCNs. GraphBTM (Zhu et al., 2018) proposed to enrich the biterm topic model (BTM) with the word co-occurrence graph encoded with GCNs. To deal with data streams, (Van Linh et al., 2020) proposed graph convolutional topic model (GCTM), which introduces a knowledge graph modeled with GCNs to the topic model. (Yang et al., 2020) presented Graph Attention TOpic Network (GATON) for correlated topic modeling. It tackles the overfitting issue in topic modeling with a generative stochastic block model (SBM) and GCNs. In contrast with these studies, we focus on integrating the topic model into GCN-based VGAE for supervised learning tasks and derive word-topic and document-topic distributions simultaneously.

2.3 Variational Graph Auto-encoders

Variational Graph Auto-encoders (VGAEs) have been widely used in graph representation learning and graph generation. The earliest study (Kipf and Welling, 2016) proposed VGAE method, which extended variational auto-encoder (VAE) on graph structure data for learning graph embedding. Based on VGAE, (Pan et al., 2018) introduced an adversarial training to regularize the latent variables and further proposed adversarially regularized variational graph autoencoder (ARVGA). (Hasanzadeh et al., 2019) incorporated semi-implicit hierarchical variational distribution into VGAE (SIG-VAE) to improve the representation power of node embeddings. (Grover et al., 2019) proposed Graphite model that integrated an iterative graph refinement strategy into VGAE, inspired by low-rank approximations. However, to the best of our knowledge, our model is the first effort to apply VGAE to unify the topic learning and graph embedding for text classification, thus can provide better interpretability and overall performance.

3 Method

3.1 Graph Construction

Formally, we denote a corpus as C, which contains D documents and the ground truth labels $Y \in c = \{1, ..., M\}$ of documents, where M is the total number of classes in the corpus. Each document $t \in C$ is represented by a sequence of words $t = \{w_1, ..., w_{n_t}\}(w_i \in v)$, where n_t is the number of words in document t and v is the vocabulary of size V.

From the whole corpus, we build a word correlation graph $G = (v, e)$ containing word nodes v and edges e, to capture the word co-occurrence information. Similar to previous work (Yao et al., 2019), we utilize the positive point mutual information (PPMI) to calculate the correlation between two word nodes. Formally, for two words (w_i, w_j), we have

$$PPMI(w_i, w_j) = \max(log\frac{p(w_i, w_j)}{p(w_i)p(w_j)}, 0) \quad (1)$$

where $p(w_i, w_j)$ is the probability that (w_i, w_j) co-occur in the sliding window and $p(w_i), p(w_j)$ are the probabilities of words w_i and w_j in the sliding window. They can be empirically estimated as $P(w_i, w_j) = \frac{n(w_i, w_j)}{n}$ and $P(w_i) = \frac{n(w_i)}{n}$, where $n(w_i, w_j)$ is the number of co-occurrences of (w_i, w_j) in the sliding windows, $n(w_i)$ is the number of occurrences of w_i in the sliding windows and n the total number of sliding windows. For two word nodes (w_i, w_j), the weight of the edge between them can be defined as:

$$A^v_{i,j} = \begin{cases} PPMI(w_i, w_j), & i \neq j \\ 1, & i = j \end{cases} \quad (2)$$

where $A^v \in \mathbb{R}^{V*V}$ is the adjacency matrix which represents the word correlation graph structure G.

Different from the existing studies (Yao et al., 2019; Liu et al., 2020; Wang et al., 2020) that consider all documents and words in a heterogeneous graph, we propose to build a separate graph for each document to enable inductive learning. Typically, documents can be represented by the document-word matrix $A^d \in \mathbb{R}^{D \times V}$, in which the row $A^d_i = \{x_{i1}, ..., x_{iv}\} \in \mathbb{R}^{1 \times V}$ represents the document i, and x_{ij} is the TF-IDF weight of the word j in document i. The decoupling of documents from a global pre-defined graph enables our method to handle new documents.

3.2 Topic Variational Graph Auto-encoder

Based on A^v and A^d, we propose the T-VGAE model, as shown in Figure 1. It is a deep generative model with structured latent variables based on GCNs.

3.2.1 Generative Modeling

We consider that the word co-occurrence graph A^v and the bipartite graph A^d_t of each document t are generated from the random process with two latent

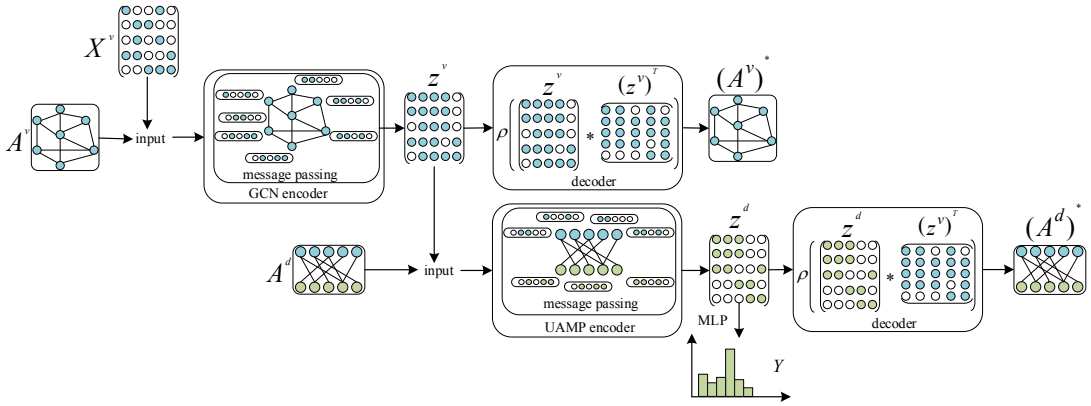
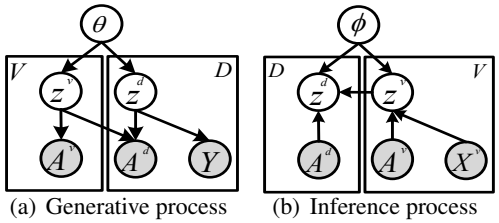


Figure 1: The architecture of T-VGAE.As shown in the Figure, for a new test document i, its latent representation z_i^d is generated by the UAMP probabilistic encoder based on its document-word vector A_i^d and learned word topic distribution matrix z^v. Then, z_i^d is fed into the trained MLP classifier f_y to predict the output label. Therefore, new test documents can be classified do not need to be included in the training process, thus enabling inductive learning of our model.

variables $z^v \in \mathbb{R}^{V\times K}$ and $z_t^d \in \mathbb{R}^{1\times K}$, where K denotes the number of latent topics. The generating process for A^v, A^d and Y are as follows (see Figure 2(a)):



(a) Generative process (b) Inference process

Figure 2: The generative and inference processes.

1. For each word i in vocabulary v, draw the latent variable z_i^v from the prior $p_\theta(z_i^v)$
2. For each observed edge $A_{i,j}^v$ between words i and j, draw $A_{i,j}^v$ from conditional distribution $p_\theta(A_{i,j}^v|z_i^v, z_j^v)$
3. For each document t in corpus C:
   (a) Draw the latent variable z_t^d from the prior $p_\theta(z_t^d)$
   (b) Draw A_t^d from the conditional distribution $p_\theta(A_t^d|z_t^d, z^v)$
   (c) Draw Y_t from the conditional distribution $p_\theta(Y_t|z_t^d)$

where θ is the set of parameters for all prior distributions. Here, we consider the centered isotropic multivariate Gaussian priors $p(z^v) = \prod_{i=1}^V p(z_i^v) = \prod_{i=1}^V \mathcal{N}(z_i^v|0, I)$ and $p(z^d) = \prod_{t=1}^D p(z_t^d) = \prod_{t=1}^D \mathcal{N}(z_t^d|0, I)$.

Notice that the priors $p(z^v)$ and $p(z^d)$ are parameter free in this case. According to the above generative process, we can maximize the marginal likelihood of observed graph A^v, A^d and Y to learn parameters θ and latent variables as follows:

$$\begin{aligned} &p(A^v, A^d, Y|Z^v, Z^d, X^v) = \\ &\prod_{t=1}^{D} p_\theta(Y_t|z_t^d) p_\theta(A_t^d|z_t^d, z^v) p_\theta(z_t^d) \\ &\prod_{i=1}^{V}\prod_{j=1}^{V} p_\theta(A_{i,j}^v|z_i^v (z_j^v)^\top) p_\theta(z^v) \end{aligned} \tag{3}$$

Because the inference of true posterior of latent variable z^v and z^d is intractable, we further introduce the variational posterior distribution $q_\phi(z^v, z^d|A^d, A^v, X^v)$ with parameters ϕ to approximate the true posterior $p_\theta(z^v, z^d) = p_\theta(z^v)p_\theta(z^d)$. We make the structured mean-field (SMF) assumption $q_\phi(z^v, z^d|A^d, A^v, X^v) = q_\phi(z^v|A^v, X^v)q_\phi(z^d|A^d, z^v)$, where $X^v \in \mathbb{R}^{V\times M}$ are the feature vectors of words and M is the dimension of the feature vectors (see Figure 2(b)). We can yield the following tractable stochastic evidence lower bound (ELBO):

$$\begin{aligned} &\mathcal{L}(\theta, \phi; A^v, A^d, X^v) \\ &= \mathbb{E}_{q_\phi(z^v|A^v, X^v)}[\log p_\theta(A^v|z^v)] \\ &+ \mathbb{E}_{q_\phi(z^d|A^d, z^v)}[\log p_\theta(A^d|z^d, z^v)] \\ &+ \mathbb{E}_{q_\phi(z^d|A^d, z^v)}[\log p_\theta(Y|z^d)] \\ &- KL[q_\phi(z^v|A^v, X^v)||p_\theta(z^v)] \\ &- KL[q_\phi(z^d|A^d, z^v)||p_\theta(z^d)] \end{aligned} \tag{4}$$

where the first three terms are the reconstruction terms, and the latter two terms are the Kullback-Leibler (KL) divergences of variational posterior

distributions and true posterior distributions. Using auto-encoding variational Bayes (AVB) approach (Kingma and Welling, 2013), we are able to parametrize the variational posteriors q_ϕ and true posteriors p_θ with the GCN-based probabilistic encoder and decoder, to conduct neural variational inference (NVI).

3.2.2 Graph Convolutional Probabilistic Encoder

For the latent variable z^v, we make the mean-field approximation that: $q_\phi(z^v|A^v, X^v) = \prod_{i=1}^{V} q_\phi(z_i^v|A^v, X^v)$. For simplify the model inference, we consider the multivariate normal variational posterior with a diagonal covariance matrix as previous neural topic models (Miao et al., 2016; Bai et al., 2018)that: $q_\phi(z_i^v|A^v, X^v) = \mathcal{N}(z_i^v|\mu_i^v, diag((\sigma_i^v)^2))$, where $\mu_i^v, (\sigma_i^v)^2$ are the mean and diagonal covariance of the multivariate Gaussian distribution.

We use the graph convolutional neural network to parametrize the above posterior and inference z^v with the input graph A^v and feature vectors X^v:

$$\begin{aligned}(H^v)^{l+1} &= \rho(\hat{A}^v(H^v)^l(W^v)^l)\\ \mu^v &= \rho(\hat{A}^v(H^v)^{l+1}(W_\mu^v)^{l+1})\\ log\sigma^v &= \rho(\hat{A}^v(H^v)^{l+1}(W_\sigma^v)^{l+1})\end{aligned} \tag{5}$$

where μ^v, σ^v are matrices of μ_i^v, σ_i^v, l is the number of GCN layers, we use one layer in our experiments, $\{W_\mu^v, W_\sigma^v\} \in \phi$ are weight matrices, ρ is the ReLU, $\hat{A}^v = (D^v)^{-\frac{1}{2}}A^v(D^v)^{-\frac{1}{2}}$ is the symmetrically normalized adjacent matrix of the word graph, and D^v denotes the corresponding degree matrix. The input of GCN is the feature vectors X_v which is initialized as the identity matrix I, i.e., $(H^v)^0 = X^v = I$, same as in (Yao et al., 2019). Then, z^v can be naturally sampled as follows according to the reparameterization trick (Kingma and Welling, 2013): $z^v = \mu^v + \sigma^v \odot \epsilon$, where $\odot$ is the element-wise product, and $\epsilon \sim \mathcal{N}(0, I)$ is the noise variable. Through the message propagation of the GCN layer, words that co-occur frequently tend to achieve similar representations in the latent topic space.

Similar to z^v, we also have:

$$\begin{aligned}q_\phi(z^d|A^d, z^v) &= \prod_{t=1}^{D} q_\phi(z_t^d|A_t^d, z^v)\\ q_\phi(z_t^d|A_t^d, z^v) &= \mathcal{N}(z_t^d|\mu_t^d, diag((\sigma_t^d)^2))\end{aligned} \tag{6}$$

where $\mu_t^d, (\sigma_t^d)^2$ are the mean and diagonal covariance of the multivariate Gaussian distribution. Although there are two types of nodes - word and document - in the bipartite graph A^d, we mainly focus on learning representations of document nodes based on the representations of word nodes learned from A^v in this step. Therefore, we propose the unidirectional message passing (UDMP) process on A^d, which propagates the information from word nodes to documents: $H_t^d = \rho(\sum_{i=1}^{V} A_{ti}^d z_i^v W^d)$ where ρ is the Relu activation function, W^d is the weight matrix.

Then, we parametrize the posterior and inference z^d based on UDMP:

$$\begin{aligned}\mu^d &= UDMP(A^d, z^v, W_\mu^d)\\ log\sigma^d &= UDMP(A^d, z^v, W_\sigma^d)\end{aligned} \tag{7}$$

where μ^d, σ^d are matrices of $\mu_t^d, (\sigma_t^d)^2$, $UDMP$ is the message passing as in Equation 4, W_μ^d, W_σ^d are weight matrices. Similarly, we sample z^d as follows $z^d = \mu^d + \sigma^d \odot \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, I)$ is the noise variable. Through the propagation mechanism of UDMP, documents which share similar words tend to yield similar representations in the latent topic space.

Although T-VGAE can learn topics z^v and document-topic representations z^d as in traditional topic models, we do not focus on proposing a novel topic model, but aim to combine the topic model with VGAE, to improve word and document representations with latent topic semantic and provide probabilistic interpretability. Moreover, rather than learning topics and document-topic representations from the document-word feature A^d as LDA topic models (Blei et al., 2003), we propose to learn word-topic representations z^v from word co-occurrence matrix A^v, and then infer document-topic representations z^d based on the document-word feature A^d and word-topic representations z^v, which is similar to the Biterm topic model (Yan et al., 2013).

3.2.3 Probabilistic Decoder

With the learned z^v and z^d, ideally, the observed graph A^v and A^d can be reconstructed through a decoding process. For A^v, we assume $P_\theta(A^v|z^v)$ conforms to a multivariate Gaussian distribution, whose mean parameters are generated from the inner product of the latent variable z^v:

$$\begin{aligned}P_\theta(A^v|z^v) &= \prod_{i=1}^{V} p_\theta(A_i^v|z^v)\\ p_\theta(A_i^v|z^v) &= \prod_{i=1}^{V} \mathcal{N}(A_i^v|\rho(z_i^v(z^v)^\top), I)\end{aligned} \tag{8}$$

where ρ is the nonlinear activation function.

Similarly, the inner product between z^v and z^d is used to generate A^d, which is sampled from the multivariate Gaussian distribution:

$$P_\theta(A^d|z^d, z^v) = \prod_{i=1}^{D} p_\theta(A_i^d|z_i^d, z^v)$$
$$P_\theta(A_i^d|z_i^d, z^v) = \prod_{i=1}^{D} \mathcal{N}(A_i^d|\rho(z_i^d(z^v)^\top), I) \quad (9)$$

For categorical labels Y, we assume $p_\theta(Y|z^d)$ follows a multinomial distribution $P_\theta(Y|z^d) = Mul(Y|f_y(z^d))$, whose label probability vectors are generated from z^d, where f_y is the multi-layer neural network. For each document t, the prediction is given by $\hat{y}_t = \underset{y \in c}{argmax}\, P_\theta(y|f_y(z_t^d))$.

3.2.4 Optimization

We can rewrite Equation 4 to yield the final variational objective function:

$$\begin{aligned}\mathcal{L}(\theta, \phi) \approx & \sum_{i=1}^{V}\sum_{j=1}^{V} \log p_\theta(A_{i,j}^v|z_i^v, z_j^v) \\ & + \sum_{t=1}^{D}\Big(\log p_\theta(A_t^d|z_t^d, z^v) + \log p_\theta(Y_t|z_t^d)\Big) \\ & - KL[q_{\phi(z^v)}||p_\theta(z^v)] \\ & - KL[q_{\phi(z^d)}||p_\theta(z^d)]\end{aligned} \quad (10)$$

with following reconstruction terms and KL divergences:

$$\begin{aligned}\log p_\theta(A_i^v|z^v) &\approx ||A_i^v - \rho(z_i^v(z^v)^\top)||^2 \\ \log p_\theta(A_t^d|z_t^d, z^v) &\approx ||A_t^d - \rho(z_t^d(z^v)^\top)||^2 \\ \log p_\theta(Y_t|z_t^d) &\approx Y_t \log \hat{y}_t + (1 - Y_t)\log(1 - \hat{y}_t) \\ KL[q_\phi(z_i^v)||p_\theta(z_i^v)] & \\ \approx \frac{1}{2}\sum_{j=1}^{V}&((\mu_{ij}^v)^2 + (\sigma_{ij}^v)^2 - (1 + \log(\sigma_{ij}^v)^2)) \\ KL[q_\phi(z_t^d)||p_\theta(z_t^d)] & \\ \approx \frac{1}{2}\sum_{j=1}^{V}&((\mu_{tj}^d)^2 + (\sigma_{tj}^d)^2 - (1 + \log(\sigma_{tj}^d)^2))\end{aligned} \quad (11)$$

Through maximizing the objective with stochastic gradient descent, we jointly learn the latent word and document representations, which can efficiently reconstruct observed graphs and predict ground truth labels.

4 Experiment

In this section, to evaluate the effectiveness of our proposed T-VGAE, experiments are conducted on both supervised and semi-supervised text classification tasks, as well as unsupervised topic modeling tasks.

Table 2: Summary statistics of five datasets (Yao et al., 2019)

Dataset	Doc	Train	Test	Word	Node	Class	Average Len
20NG	18,846	11,314	7,532	42,757	42,757	20	221.26
R8	7,674	5,485	2,189	7,688	7,688	8	65.72
R52	9,100	6,532	2,568	8,892	8,892	52	69.82
Ohsumed	7,400	3,357	4,043	14,157	14,157	23	135.82
MR	10,662	7,108	3,554	18,764	18,764	2	20.39

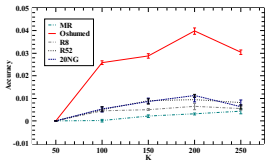


Figure 3: The augmentation of test accuracy with our model under different topic size k.

4.1 Datasets and settings

4.1.1 Datasets

We conduct experiments on five commonly used text classification datasets: 20NewsGroups, Ohsumed, R52 and R8, and MR. We use the same data preprocessing as in (Yao et al., 2019). The overview of the five datasets is depicted in Table 2.

4.1.2 Baselines

We compare our method with the following two categories of baselines:

text classification: 1)TF-IDF+LR: the classical logistic regression method based on TF-IDF features. 2) CNN (Kim, 2014): the convolutional neural network based method with pre-trained word embeddings. 3) LSTM (Liu et al., 2016): the LSTM based method with pre-trained word embeddings. 4) SWEM (Shen et al., 2018): the word embedding model with pooling strategies. 5) fastText (Joulin et al., 2016): the averages word embeddings for text classification. 6) Graph-CNN (Peng et al., 2018a): a graph CNN model based on word embedding similarity graphs 7) LEAM (Wang et al., 2018): the label-embedding attentive models with document embeddings based on word and label descriptions. 8) TextGCN (Yao et al., 2019): a GCN model with a corpus-level graph to learn word and document embeddings. 9) DHTG (Wang et al., 2020): a GCN model with a dynamic hierarchical topic graph based on the topic model.

topic modeling: 1) LDA (Blei et al., 2003): a classical probabilistic topic model. 2) NVDM

[1]Its code is not released yet, therefore we only report the test micro precision here.

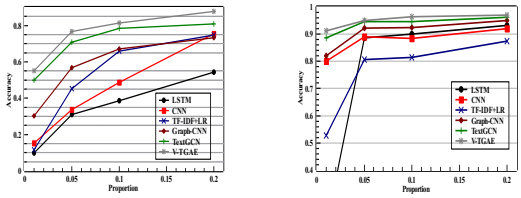
Table 3: Micro precision, recall and F1-Score on document classification task. We report mean ± standard deviation averaged on 10 times following previous methods (Yao et al., 2019).

Model	**20NG**			**MR**			**Ohsumed**	
Measure	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
TF-IDF+LR	0.8212 ± 0.0000	0.8301± 0.0000	0.8300± 0.0000	0.7452 ± 0.0000	0.7432 ± 0.0000	0.7431 ± 0.0000	0.5454 ± 0.0000	0.5454 ± 0.0000
CNN	0.8213 ± 0.0052	0.7844 ± 0.0022	0.7880± 0.0020	0.7769 ± 0.0007	0.7366 ± 0.0026	0.7390 ± 0.0018	0.5842 ± 0.0106	0.4429 ± 0.0057
LSTM	0.7321 ± 0.0185	0.7025 ± 0.0046	0.7016 ± 0.0050	0.7769 ± 0.0086	0.7526 ± 0.0062	0.7432 ± 0.0024	0.4925 ± 0.0107	0.4852 ± 0.0046
SWEM	0.8518 ± 0.0029	0.8324 ± 0.0016	0.8273 ± 0.0021	0.7668 ± 0.0063	0.7481 ± 0.0026	0.7428 ± 0.0023	0.6313 ± 0.0055	0.6280 ± 0.0041
LEAM	0.8190 ± 0.0024	0.8026 ± 0.0014	0.8132 ± 0.0021	0.7693 ± 0.0045	0.7438 ± 0.0036	0.7562 ± 0.0023	0.5859 ± 0.0079	0.5832 ± 0.0026
fastText	0.7937 ± 0.0030	0.7726 ± 0.0046	0.7730 ± 0.0028	0.7512 ± 0.0020	0.7411 ± 0.0013	0.7406 ± 0.0025	0.5769 ± 0.0049	0.5594 ± 0.0012
Graph-CNN	0.8139 ± 0.0032	0.8106 ± 0.0056	0.8099 ± 0.0042	0.7721 ± 0.0027	0.7643 ± 0.0034	0.7667 ± 0.0029	0.6390 ± 0.0053	0.6345 ± 0.0032
TextGCN	0.8634 ± 0.0009	0.8627 ± 0.0006	0.8627 ± 0.0011	0.7673 ± 0.0020	0.7640 ± 0.0010	0.7636 ± 0.0010	0.6834 ± 0.0056	0.6820 ± 0.0014
DHTG [1]	0.8713 ± 0.0007	-	-	0.7721 ± 0.0011	-	-	0.6880 ± 0.0033	-
T-VGAE	**0.8808** ± 0.0006	**0.8804** ± 0.0010	**0.8802** ± 0.0009	**0.7803** ± 0.0011	**0.7805** ± 0.0011	**0.7805** ± 0.0011	**0.7002** ± 0.0014	**0.7008** ± 0.0010

Model	**Ohsumed**	**R52**			**R8**		
Measure	F1	Precision	Recall	F1	Precision	Recall	F1
TF-IDF+LR	0.5453 ± 0.0000	0.8693 ± 0.0000	0.8670 ± 0.0000	0.8687 ± 0.0000	0.9375 ± 0.0000	0.9366 ± 0.0000	0.9344 ± 0.0000
CNN	0.4295 ± 0.0018	0.8760 ± 0.0048	0.8711± 0.0012	0.8431 ± 0.0015	0.9572± 0.0052	0.9534± 0.0014	0.9519± 0.0017
LSTM	0.4864 ± 0.0060	0.9053 ± 0.0091	0.8932 ± 0.0022	0.8910 ± 0.0018	0.9634 ± 0.0033	0.9612 ± 0.0025	0.9608 ± 0.0031
SWEM	0.6252 ± 0.0032	0.9295 ± 0.0024	0.9236 ± 0.0022	0.9180 ± 0.0022	0.9531 ± 0.0026	0.9487 ± 0.0024	0.9462 ± 0.0018
LEAM	0.5824 ± 0.0022	0.9183 ± 0.0023	0.9041 ± 0.0017	0.9002 ± 0.0030	0.9330 ± 0.0024	0.9211 ± 0.0012	0.9207 ± 0.0014
fastText	0.5587 ± 0.0026	0.9282 ± 0.0009	0.9146 ± 0.0012	0.9112 ± 0.0026	0.9611 ± 0.0021	0.9467 ± 0.0018	0.9501 ± 0.0022
Graph-CNN	0.6278 ± 0.0023	0.9274 ± 0.0023	0.9106 ± 0.0030	0.9098 ± 0.0028	0.9697 ± 0.0012	0.9387 ± 0.0018	0.9403 ± 0.0014
TextGCN	0.6820 ± 0.0012	0.9354 ± 0.0018	0.9340 ± 0.0012	0.9339 ± 0.0010	0.9704 ± 0.0010	0.9703 ± 0.0009	0.9700 ± 0.0012
DHTG	-	0.9393 ± 0.0010	-	-	0.9733 ± 0.0006	-	-
T-VGAE	**0.7004± 0.0010**	**0.9505** ± 0.0010	**0.9500** ± 0.0012	**0.9500** ± 0.0010	**0.9768** ± 0.0014	**0.9766** ± 0.0009	**0.9765** ± 0.0009

Table 4: Test Accuracy on document classification task averaged on 10 times using different layers of GCN encoder, i.e. $l \in (0, 1, 2, 3)$.

Model	**R52**	**R8**
$l = 0$	0.9143 ± 0.0015	0.9495 ± 0.0011
$l = 1$	**0.9505** ± 0.0010	**0.9768** ± 0.0014
$l = 2$	0.8942 ± 0.0012	0.9667 ± 0.0014
$l = 3$	0.7326 ± 0.0012	0.8795 ± 0.0010



(a) 20NG (b) R8

Figure 4: Test accuracy of different models under varying training data proportions.

(Miao et al., 2016): a deep neural variational document topic model. 3) AVITM (Srivastava and Sutton, 2017): an autoencoding variational Bayes (AEVB) topic model based on LDA. 4) GraphBTM (Zhu et al., 2018): an enriched biterm topic model (BTM) with the word co-occurrence graph encoded by GCN.

4.1.3 Settings

Following (Yao et al., 2019), we set the hidden size K of latent variables and other neural network layers as 200 and set the window size in PPMI as 20. The dropout is only utilized in the classifier, and is set to 0.85. We train our model for a maximum of 1000 epochs with Adam (Kingma and Ba, 2015) under learning rate 0.05. 10% of the data set is randomly sampled and spared as the validation set for model selection. The parameter settings of all baselines are the same as their original papers or implementations.

4.2 Performance

4.2.1 Supervised Classification

We present the test performances of models in text classification among five datasets in Table 3. We can see that our model consistently outperforms all the baselines on each dataset, which proves the effectiveness of our proposed methods. Compared with TextGCN, our method yields better performance in both datasets. It demonstrates the importance of integrating the latent semantic structures in text classification. It is also observed from the superior performance of DHTG when compared with TextGCN. However, DHTG only learns from the document-word correlation while our method fully exploits both word-word and document-word correlation information, resulting in a significant improvement over DHTG. This proves the effectiveness of unified topic modeling and graph representation learning in text classification. Moreover, there are no test documents involved during the training of our method, which shows the inductive learning ability of our method, different from TextGCN and DHTG which requires a global graph including all documents and words.

4.2.2 Effects of Correlation Information of Different Order

In Table 4, we further present the test accuracy of our method using different layers of GCN en-

coder, to demonstrate the impact of a different order of word-word correlation information in A^v. On datasets R52 and R8, our method achieves the best performance when the layer number is 1. This is different from TextGCN and DHTG, which generally have the best performance with 2 layer GCN. A possible reason is that our model has already considered one-hop document-word relation information when encoding document-word graph A^d. If the layer number is set to 1 when encoding A^v, it actually integrates two-hop neighborhood information, thus achieves a similar effect to TextGCN and DHTG. In Table 4, we further present the test accuracy of our method using different layers of GCN encoder, to demonstrate the impact of different orders of word-word correlation information in A^v. On datasets R52 and R8, our method achieves the best performance when the layer number is 1. This is different from TextGCN and DHTG, which generally have the best performance with 2 layer GCN. A possible reason is that our model has already considered one-hop document-word relation information when encoding document-word graph A^d. If the layer number is set to 1 when encoding A^v, it actually integrates two-hop neighborhood information, thus achieves a similar effect to TextGCN and DHTG.

4.2.3 Effects of Number of Topics

Figure 3 shows the changes of the test accuracy along with different numbers of topics on five datasets. We can see that the test accuracy on five datasets generally improves with the increase of the number of topics and reaches the peak when the topic number is around 200. The number of topics shows more impact on the Oshumed dataset than on the other four datasets. This does not seem to be related to the number of classes in the dataset. We suspect it has to do with the nature of the text (medical domain vs. other domains).

4.2.4 Semi-Supervised Classification

In Figure 4, we further present the semi-supervised classification test accuracy on datasets 20NG and R8 where different proportions (1%, 5%,10% and 20%) of the original training set are used. We can see that, in cases where labeled samples are limited, our model still consistently outperforms all the baselines. Compared with other methods, TextGCN and our model can preserve good performance with few labeled samples (1%, 5%). This illustrates the effect of label propagation in GCN for semi-supervised learning. When compared with TextGCN, our model yields better performance because of its inductive learning capability and the incorporation of the latent topic semantics.

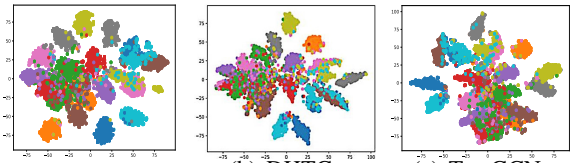
4.2.5 Document Topic Modelling

Table 5: The top-10 words and coherence score of topics in 20NG dataset from z^v.

Category	Topic
Sport	T57: team season hockey game nhl players win play baseball chip 1.8902 T64: clipper hockey season team encryption key nhl toal baseball gt 1.1985
Autos	T61: lcs x11r5 xpert x 6128 cars enterpoop lintlibdir car xwininfo 1.1931 T62: x11r5 x car cars lcs encryption daubenspeck xterm clipper xpert 0.8977
Elec	T12: mac centris graphics quadra iisi apple c650 tomj geb powerbook 1.1603 T71: mac dod quadra centris apple bike iisi encryption lciii lc 0.9789

Table 6: The average topic coherence (higher is better) and perplexity (lower is better) with different topic numbers.

Metrics	Topic coherence		Perplexity	
Model	**K=50**	**K=200**	**K=50**	**K=200**
LDA	0.17	0.14	728	688
NVDM	0.08	0.06	837	884
AVITM	0.24	0.19	1059	1128
GraphBTM	0.28	0.26	-	-
T-VGAE	**0.37**	**0.59**	**615**	**665**



(a) T-VGAE (b) DHTG (c) TextGCN

Figure 5: The t-SNE visualization of test document embeddings of 20NG by different models.

We further evaluate the performance of models on unsupervised topic modeling tasks. We generally assume that the more topics are coherent, the more they are interpretable. Following (Srivastava and Sutton, 2017), We use the average pairwise PMI of the top 10 words in each topic and the perplexity with the ELBO as quality measures of topics. We show in Table 6 the measures under different topic numbers in the 20NG dataset. We remove the supervised loss of our method and the result of GraphBTM is not presented for unable to learn document topic representation for each document.

In the table, we can see that our model outperforms the others in terms of topic coherence, which could be attributed to the combination of word co-occurrence graph and message passing in GCN. The message passing leads to similar representations of words that co-occur frequently in the latent topic space, thus improves the semantic coherence of learned topics, as shown in Table 5 that related words tend to belong to the same topic. Our method also benefits from document-word correlation, and yield better performance when compared with GraphBTM which encode bi-term graph via GCN.

4.2.6 Document Representations

We utilize t-SNE to visualize the latent test document representations of the 20NG dataset learned by our model, DHTG and TextGCN in Figure 5, in which each dot represents a document and each color represents a category. Our method yields the best clustering results compared with the others, which means the topics are more consistent with pre-defined classes. It shows the superior interpretability of our method for modeling the latent topics along with both word co-occurrence graph and document-word graph when compared with DHTG.

5 Conclusion

In this paper, we proposed a novel deep latent variable model T-VGAE via combining the topic model with VGAE. It can learn more interpretable representations and leverage the latent topic semantic to improve the classification performance. T-VGAE inherits advantages from the topic model and VGAE: probabilistic interpretability and efficient label propagation mechanism. Experimental results demonstrate the effectiveness of our method along with inductive learning. As future work, it would be interesting to explore better-suited prior distribution in the generative process. It is also possible to extend our model to other tasks, such as information recommendation and link prediction.

Acknowledgments

This research is supported by the CSC Scholarship offered by China Scholarship Council. We would like to thank the anonymous reviewers for their constructive comments. We thank MindSpore for the partial support of this work, which is a new deep learning computing framework[1].

References

Haoli Bai, Zhuangbin Chen, Michael R Lyu, Irwin King, and Zenglin Xu. 2018. Neural relational topic models for scientific article analysis. In *CIKM*, pages 27–36.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *JMLR*, (Jan):993–1022.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4927–4936.

Aditya Grover, Aaron Zweig, and Stefano Ermon. 2019. Graphite: Iterative generative modeling of graphs. In *ICML*, pages 2434–2444.

Arman Hasanzadeh, Ehsan Hajiramezanali, Krishna Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. 2019. Semi-implicit graph variational auto-encoders. In *NIPs*.

Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text level graph neural network for text classification. *arXiv preprint arXiv:1910.02356*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv:1607.01759*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

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

DP Kingma and JL Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv:1611.07308*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *IJCAI*, pages 2873–2879. AAAI Press.

Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. *AAAI*.

[1] https://www.mindspore.cn/

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *ICML*, pages 1727–1736.

Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. 2018. Adversarially regularized graph autoencoder for graph embedding. In *IJCAI*, pages 2609–2615.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018a. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *WWW*, pages 1063–1072.

Min Peng, Qianqian Xie, Yanchun Zhang, Hua Wang, Xiuzhen Jenny Zhang, Jimin Huang, and Gang Tian. 2018b. Neural sparse topical coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2332–2340.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv:1805.09843*.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv:1703.01488*.

Ngo Van Linh, Tran Xuan Bach, and Khoat Than. 2020. Graph convolutional topic model for data streams. *arXiv*, pages arXiv–2003.

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *ACL*, pages 3308–3318.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. *arXiv:1805.04174*.

Zhengjue Wang, Chaojie Wang, Hao Zhang, Zhibin Duan, Mingyuan Zhou, and Bo Chen. 2020. Learning dynamic hierarchical topic graph with graph convolutional network for document classification.

Qianqian Xie, Jimin Huang, Pan Du, Min Peng, and Jian-Yun Nie. 2021. Graph topic neural network for document representation. In *Proceedings of The Web Conference 2021*.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1445–1456.

Liang Yang, Fan Wu, Junhua Gu, Chuan Wang, Xiaochun Cao, Di Jin, and Yuanfang Guo. 2020. Graph attention topic modeling network. In *WWW*, pages 144–154.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *AAAI*.

Haopeng Zhang and Jiawei Zhang. 2020. Text graph transformer for document classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8322–8327.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every document owns its structure: Inductive text classification via graph neural networks. *arXiv preprint arXiv:2004.13826*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*, pages 2205–2215.

Qile Zhu, Zheng Feng, and Xiaolin Li. 2018. Graphbtm: Graph enhanced autoencoded variational inference for biterm topic model. In *EMNLP*, pages 4663–4672.