AS-GCN: Adaptive Semantic Architecture of Graph Convolutional Networks for Text-Rich Networks

Zhizhi Yu[1], Di Jin[1], Ziyang Liu[2], Dongxiao He[*1], Xiao Wang[*3], Hanghang Tong[4], and Jiawei Han[4]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]School of Software, Tsinghua University, Beijing, China
[3]School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China
[4]Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, IL, USA
[1]{yuzhizhi, jindi, hedongxiao}@tju.edu.cn, [2]liu-zy21@mails.tsinghua.edu.cn
[3]xiaowang@bupt.edu.cn, [4]{htong, hanj}@illinois.edu

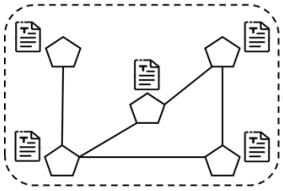
***Abstract*—Graph Neural Networks (GNNs) have demonstrated great power in many network analytical tasks. However, graphs (i.e., networks) in the real world are usually text-rich, implying that valuable semantic structure information needs to be carefully considered. Existing GNNs for text-rich networks typically treat text as attribute words alone, which inevitably leads to the loss of important semantic structure information, limiting the representation capability of GNNs. In this paper, we propose an end-to-end adaptive semantic architecture of graph convolutional networks, namely AS-GCN, which unifies neural topic model and graph convolutional networks, for text-rich network representation. Specifically, we utilize a neural topic model to extract the global topic semantics, and accordingly augment the original text-rich network into a tri-typed heterogeneous network, capturing both the local word sequence semantic structure and the global topic semantic structure from text. We then design an effective semantic-aware propagation of information by introducing a discriminative convolution mechanism. We further propose two strategies, that is, distribution sharing and joint training, to adaptively generate a proper network structure based on the learning objective to improve network representation. Extensive experiments on text-rich networks illustrate that our new architecture outperforms the state-of-the-art methods by a significant improvement. Meanwhile, this architecture can also be applied to e-commerce search scenes, and experiments on a real e-commerce problem from JD further demonstrate the superiority of the proposed architecture over the baselines.**

***Index Terms*—graph neural networks, adaptive semantic architecture, text-rich networks**

I. Introduction

Networks are ubiquitously used to represent data in a wide range of fields, including social network analysis, bioinformatics, and computer network security. With their prevalence, it is particularly important to learn effective representations of networks and apply them to downstream tasks. Recently, Graph Neural Networks (GNNs) [1], [2], a class of neural networks designed to learn network data, have shown remarkable success in capturing network representation, and have been widely applied in tackling network analytical tasks, such as node classification [3], link prediction [4], and recommendation [5].

*Corresponding author.



ID	Document Text
1	Privacy preservation in wireless sensor…
2	Tabu search for the Steiner problem…
3	Feature interaction: a critical review…
4	A study of malware in P2P networks…
5	Understanding the paradoxical effects…

Fig. 1: An illustrative example of a text-rich network constructed from DBLP [13], where each node is associated with textual description composed of its title and abstract.

The typical GNNs [6], [7] and their variants [8], [9] usually follow a message-passing design, i.e., obtain network representation through the propagation and aggregation of attributes over network topology. However, networks in the real world are usually text-rich [10], [11] (see an example in Figure 1), where the text corresponding to each node is not only the collection of attribute words, but also contains valuable context information of word sequence (e.g., attribute words "privacy" and "preservation" are semantically relevant, rather than independent), and reflects the semantic structure of the topic (e.g., in two documents without many common attribute words, their semantics may also be interrelated). As a result, it is difficult for the existing GNNs [12], which only treat text as independent attribute words, to fully utilize richer structural semantics of text to perform propagation, significantly limiting the representation capability of GNNs. Therefore, it is imperative to explore a new architecture of GNNs which can fully embody the most informative semantic structure information of text, so that can be effectively applied to text-rich network representation.

Nevertheless, it is technically challenging to effectively design a new architecture of GNNs for text-rich network representation. Particularly, two obstacles need to be addressed. First, semantic structure information of text should be taken into consideration, including the local word sequence structure and the global topic structure. Considering these information fundamentally drives the learned architecture to maintain more semantics and be more robust to noise in real observations.

Meanwhile, by fully utilizing text semantics, the architecture itself could alleviate the homophily assumption [14], [15] of GNNs, and achieve the optimal balance between topology and attributes (i.e., learn different contributions of these two parts automatically informed by the ground truth). Unfortunately, the majority of current GNNs [16] do not account for the richer structural semantics of text, thus inevitably limiting their performance. Second, the network architecture itself should be dynamically adaptive. Learning semantic structure information from text inevitably leads to bias and uncertainty. Therefore, it makes sense that the confidence of a network architecture would be greater if this architecture is dynamically estimated aiming at the given learning objectives. Therefore, an optimal network architecture ought to make allowance for the ground truth such as node classification, which is particularly significant while ignored by the existing GNNs [17].

To address these aforementioned issues, in this paper, we propose an end-to-end adaptive semantic architecture of graph convolutional networks (GCN), i.e., AS-GCN, for text-rich network representation. As shown in Figure 2, it consists of two data-driven components, that is, a neural topic model (NTM) for extracting the global topic semantics from raw text, and a network learning module for semantic-aware propagation of information on the augmented tri-typed network. The two modules, one powered by NTM and the other by GCN, are designed to have mutually complementary inductive biases. To be specific, for NTM, we mainly introduce an encoding-decoding process which models raw text to generate topic distribution and word distribution. For network learning module, we first transform the original text-rich network into an augmented tri-typed heterogeneous network utilizing the words extracted from raw text and the distributions obtained from NTM, capturing both the local word sequence semantics and the global topic semantics. We then introduce a discriminative hierarchical convolution mechanism, based on the type of edges, to effectively aggregate information from this augmented network. Furthermore, we train these two modules, including NTM and network learning, together by leveraging distribution sharing and joint training strategies, so as to adaptively estimate an appropriate network architecture based on the learning objectives. Last but not the least, this architecture is almost orthogonal to most GNNs, and thus can be readily incorporated into various GNNs to further improve their performance.

We summarize our main contributions as follows:

- We find that, the message-passing mechanism adopted by existing GNNs fails on taking full advantage of the most informative semantic structure information in text-rich networks (since text is generally only regarded as independent attribute words); and meanwhile, they typically take the network architecture as a ground truth description of the relationship between nodes, despite that the data we employed is often imperfect.
- We propose a novel end-to-end adaptive semantic GCN architecture for text-rich network representation. To the best of our knowledge, this is the first attempt devoted to use the valuable semantic structure information of text, including the local word sequence structure and the global topic structure, for text-rich network modeling. Moreover, the architecture itself can not only effectively alleviate the homophily assumption of GNNs, but also achieve the optimal balance between topology and attributes by fully utilizing the richer structural semantics.
- Extensive experiments on four public text-rich network datasets as well as one real e-commerce application demonstrate the superiority of the proposed new approach over state-of-the-art methods.

TABLE I: Summary of notations.

Notations	Descriptions
G	A network.
V, E	The sets of nodes and edges of a network.
A, X	The adjacency matrix and node attribute matrix.
D	The node degree matrix.
R	The set of raw text.
e_{ij}	The edge between nodes v_i and v_j.
a_{ij}	The connection between nodes v_i and v_j.
x_i	The attribute vector of a given node v_i.
k	The predefined topic number.
u	The vocabulary size.
p_w	The probability of predicted words in NTM.
θ, W_r	The topic distribution and word distribution.
$p(z\|x), p(x\|z)$	The probabilities of encoding and decoding processes in NTM.
σ	The non-linear activation function.
h_i^l	The feature representation of node v_i at the l-th layer.
Φ	The set of edge-types of an augmented network.
N^ϕ	The set of neighbors connected via edge-type ϕ.
α_{ij}^ϕ	Weight of edge-type based node pair (v_i, v_j).
γ^ϕ	Weight of edge-type ϕ.
$D_{KL}(\cdot\|\|\cdot)$	Kullback–Leibler divergence.
PMI(w_i, w_j)	The PMI value between word nodes w_i and w_j.

The rest of the paper is organized as follows. Section II gives the preliminaries. Section III proposes the new GCN architecture for text-rich network representation. We conduct experiments in Section IV and introduce the application on e-commerce search in Section V. Finally, we discuss related work in Section VI and conclude in Section VII.

II. PRELIMINARIES

We first introduce the notations and define the problem of semi-supervised node classification, and then discuss GCN which serves as the base of our new architecture.

A. Notations and Problem Definition

Let $G = (R, V, E)$ be a text-rich network, where R represents the set of raw text, $V = \{v_1, \ldots, v_n\}$ is the set of n nodes, $E = \{e_{ij}\} \subseteq V \times V$ is the set of m edges. The topological structure of G can be represented by an $n \times n$ adjacency matrix $A = (a_{ij})_{n\times n}$, where $a_{ij} = 1$ if there is an edge between nodes v_i and v_j, or 0 otherwise. The nodes are described by the attribute matrix $X = (x_i)_{n\times u}$, where attributes are extracted directly from raw text R and u is the dimension of node attributes.

Given a text-rich network G, and a labeled node set V_L containing $c \ll |V|$ nodes, where each node $v_i \in V_L$ contains

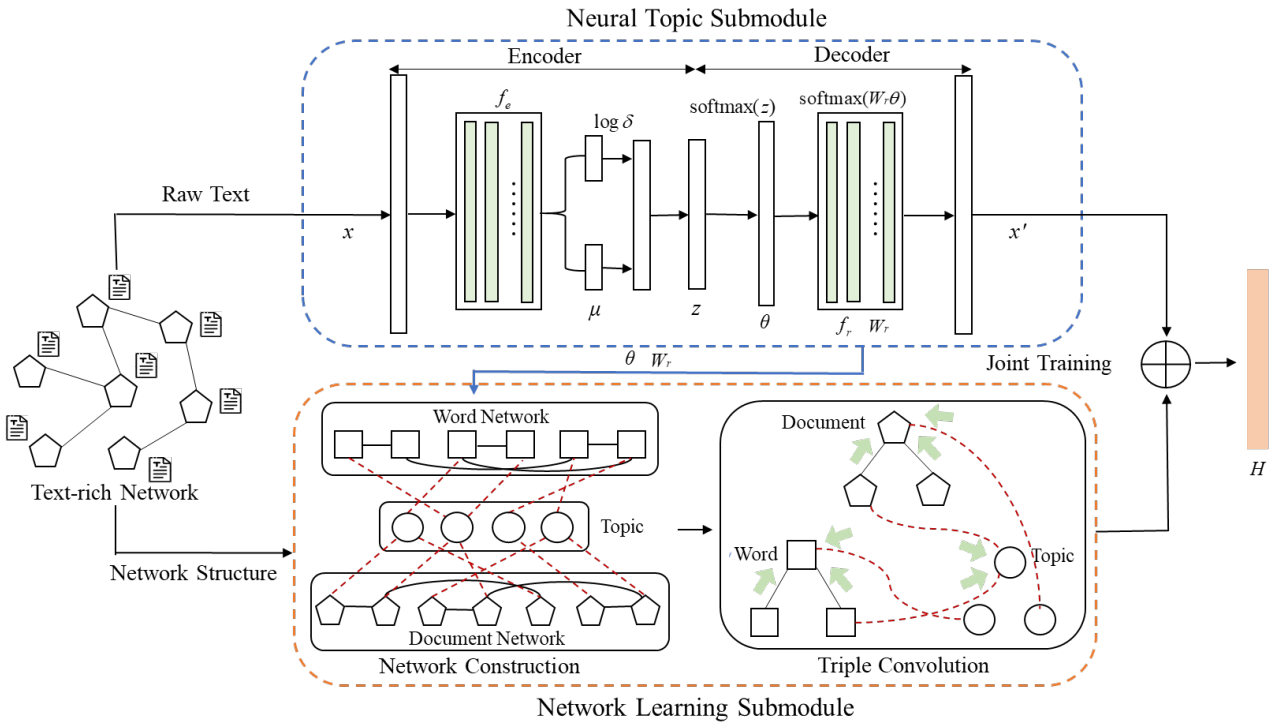


Fig. 2: The architecture of AS-GCN. It jointly trains a neural topic module for extracting the global topic semantics, and a scalable network learning module for semantic-aware propagation of information on the augmented tri-typed network.

a unique class label $y_i \in Y$. The goal of *semi-supervised node classification* is to infer the labels of nodes in $V \setminus V_L$ by learning a classification function $\mathcal{F}$. The notations we will use throughout the paper are summarized in Table I.

B. Graph Convolutional Network

Graph Convolutional Networks (GCN) [6] is a variant of multi-layer convolutional neural networks that operates directly on networks. It learns representation of each node by iteratively aggregating feature information from its topological neighbors. Mathematically, let $H^{(l)}$ be the feature representation of the l-th layer, $H^{(0)}$ be the node attribute matrix, the forward propagation can then be defined as:

$$H^{(l)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l-1)}W^{(l)}), \tag{1}$$

where $\tilde{A} = A + I$ stands for the adjacency matrix with self-loops, $\tilde{D}$ is the diagonal degree matrix of $\tilde{A}$, i.e., $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, $W^{(l)}$ is a weight matrix used to embed the given inputs (typically to a lower dimension), and σ is the non-linear activation function such as ReLU or Sigmoid. While GCN works well on several network analytical tasks [18], [19], networks in the real world are usually text-rich, it will inevitably overlook important semantic structure information if it only treats text as attribute words. This leads to the main contribution in this work, i.e., design a new adaptive semantic GCN architecture for text-rich network representation by making the most use of the semantics contained in text.

III. AS-GCN: THE PROPOSED MODEL

We first give a brief overview of the proposed method, then introduce two key components in detail, and finally discuss some optional tricks in implementation.

A. Overview

To make the architecture itself fully embody the most informative semantic structure information of text, we introduce a neural topic model to extract topic-relevant information from raw text, and accordingly design a new adaptive semantic architecture to perform discriminative convolution, so as to better learn the representations of text-rich networks. The architecture of our new approach AS-GCN is illustrated in Figure 2. It includes two main components: neural topic submodule and network learning submodule.

In neural topic module, we learn the topic distribution and word distribution through an encoding-decoding process by using raw text, for the construction of the new network structure. In network learning module, we first augment the original text-rich network into a tri-typed network utilizing the words extracted from raw text and the distributions derived from neural topic module, capturing both information from the local word sequence structure and the global topic structure. We then perform a semantic-aware propagation of information on the augmented tri-typed network through introducing a discriminative convolution mechanism, so as to distinguish and learn the contributions of network part and text part based on the learning objectives. Different from most existing works

[20], [21], our framework models both raw text and network structure, and we learn both modules in an end-to-end manner to let them mutually enhance each other.

B. Neural Topic Module

Neural topic model (NTM), which is inspired by Miao *et al.* [22] that induces latent topics in neural network, aims to learn the topic distribution of documents and word distribution of topics, so as to effectively generate a new GCN architecture on text-rich networks. NTM is based on the variational autoencoder (VAE) framework [23], and learns the latent topics through an encoding-decoding process. Specifically, let $x_i \in \mathbb{R}^u$ be the feature representation of a given node v_i, where u is the vocabulary size. Then, in the encoding process, we have:

$$\mu = f_\mu(f_e(x_i)); \log \delta = f_\delta(f_e(x_i)), \tag{2}$$

where μ and δ are the prior parameters for inducing intermediate topic distribution in the decoder, f_e, f_μ and f_δ are linear transformations with ReLU activation.

Analogous to LDA-style topic models [24], [25], the decoder can be regarded as a three-step document generation process. We first adopt Gaussian softmax [22] to draw the topic distribution of documents. Mathematically, let z be the latent topic variable, k be the predefined topic number, the topic distribution $\theta \in \mathbb{R}^k$ can be defined as:

$$z \sim \mathcal{N}(\mu, \delta^2);\ \theta = \mathrm{softmax}(z). \tag{3}$$

With the obtained topic distribution θ, the probability of predicted words $p_w \in \mathbb{R}^L$ can then be given by using a mapping function along with a non-linear transform (i.e., softmax) as:

$$p_w = \mathrm{softmax}(W_r\theta), \tag{4}$$

where W_r represents the word distribution, and $W_r^{(i,j)}$ denotes the relevance between the i-th word and j-th topic.

Finally, we draw each word from p_w to reconstruct input x_i. Considering the intermediate parameter W_r has encoded topical information, the topic representation can then be defined in the following as:

$$H_T = f_r(W_r^T), \tag{5}$$

where f_r is a linear transformation with ReLU activation.

C. Network Learning Module

We first transform the original text-rich network into an augmented tri-typed network, and then introduce a discriminative convolution mechanism, which performs a semantic-aware propagation of information on this augmented network, realizing the convolutions of topology and attributes altogether in the same system.

Network construction. To explicitly describe the local word sequence structure and the global topic structure of text, we augment the original text-rich network $G_D = (R, V_D, E_D)$ into a tri-typed heterogeneous network utilizing the words extracted from raw text and the distributions derived from NTM, as shown in the low-middle part of Figure 1. It includes three types of nodes, namely real nodes (e.g., document nodes in the original network), topic nodes (e.g., topics obtained from NTM) and entity nodes (e.g., words extracted from raw text); and four types of edges, that is, edges between real nodes such as paper citations, edges between real nodes and topic nodes reflecting the topic distribution of documents, edges between topic nodes and entity nodes representing the word distribution of topics, as well as edges between entity nodes reflecting the local word sequence semantics.

For local word sequence information, we employ a fixed size sliding window to gather co-occurrence statistics from raw text. We adopt point-wise mutual information (PMI) [26], a common measure for word associations, to construct edges between two word nodes. Mathematically, let $V_M = \{w_1, w_2, ..., w_u\}$ be the set of word nodes extracted from raw text, the PMI value between word nodes w_i and w_j can then be represented as:

$$\mathrm{PMI}(w_i, w_j) = \log \frac{s(w_i, w_j)}{s(w_i)s(w_j)}, \tag{6}$$

$$s(w_i, w_j) = \frac{\#W(w_i, w_j)}{\#W}, \tag{7}$$

$$s(w_i) = \frac{\#W(w_i)}{\#W}, \tag{8}$$

where $\#W(w_i)$ denotes the number of sliding windows containing word node w_i, $\#W(w_i, w_j)$ represents the number of sliding windows containing both word nodes w_i and w_j, and $\#W$ is the total number of sliding windows in raw text. Obviously, a positive PMI value indicates that the semantic correlation of word nodes is high, while a negative PMI value indicates low or no semantic correlation. Therefore, the set of word edges E_M can be obtained by choosing word pairs with positive PMI value.

For global topic structure information, we construct edge sets E_{DT} and E_{TM} based on the distributions obtained from NTM. Let $V_T = \{t_1, t_2, ..., t_k\}$ be the set of topic nodes, we choose topic nodes with high probability in topic distribution for each document node to build edges, and word nodes with high probability in word distribution for each topic node to generate edges.

Then, the augmented tri-typed network can be defined as:

$$G = (V_D \cup V_T \cup V_M, E_D \cup E_{DT} \cup E_{TM} \cup E_M). \tag{9}$$

Tri-typed convolution. The novel tri-typed convolution mechanism consists of two parts, including the aggregation of information from the same edge-type (i.e., intra aggregation), and the aggregation of information from different edge-types (i.e., inter aggregation). Specifically, in the intra aggregation, we adopt the same summation as GCN to aggregate the information from neighbors based on the same edge-type. Formally, let $h_i^{(l-1)}$ be the feature representation of node v_i at the (l-1)-th layer, and $h_i^{(0)}$ be the node's feature vector. Then, for each node v_i, its embedding of the edge-type ϕ at the l-th layer $h_i^{(\phi,l)}$ can be updated as:

$$h_i^{(\phi,l)} = \sum_{j \in N_i^\phi} (\widetilde{d}_i \widetilde{d}_j)^{\frac{1}{2}} h_j^{(l-1)} \quad \forall \phi \in \Phi, \tag{10}$$

where $\widetilde{d}_i$ is the degree of node v_i of augmented tri-typed network with self-loops, N_i^ϕ is the set of neighbors connected via edge-type ϕ of node v_i, and Φ is the set of edge-types.

In the inter aggregation, considering that different nodes have different edge-types, then for each node v_i, we introduce another aggregation function, i.e., concatenation $\|$, to aggregate the embeddings of different edge-types, which is defined as:

$$g_i^{(l)} = \Big\|_{\phi} h_i^{(\phi,l)}. \tag{11}$$

With the obtained $g_i^{(l)}$, the l-th layer embedding of node v_i can then be given by using a mapping function along with a non-linear transform as:

$$h_i^{(l)} = \sigma(g_i^{(l)} \cdot W^{(l)}), \tag{12}$$

where $W^{(l)}$ is the mapping matrix and σ is the non-linear activation function such as ReLU.

D. Model Training

Now we have introduced our NTM and network learning module, we are ready to put them together through an end-to-end training framework. We employ two techniques for training with semi-supervision: distribution sharing and joint training.

Distribution sharing. The performance of our network learning module depends on the quality of network structure. In our framework, we construct edges between document nodes and topic nodes, and edges between topic nodes and word nodes utilizing the distributions generated by NTM. Initially, the augmented tri-typed network is created using the distributions obtained by training NTM for 200 epochs. As the training progress, the NTM module gets improved, and could produce higher quality topic distribution and word distribution. Therefore, we share the most up-to-date distributions from the NTM with the network learning module. We have found that such a distribution sharing mechanism is beneficial for improving the performance of GCN on text-rich network representation.

Joint training. Since the augmented tri-typed network is generated partially utilizing the distributions from the NTM module, we leverage joint training to let them mutually enhance each other. For the NTM, the objective function is defined as the negative evidence lower bound, which is written as follows:

$$\mathcal{L}_{NTM} = D_{KL}(q(z)||p(z|x)) - \mathbb{E}_{q(z)}[p(x|z)], \tag{13}$$

where the first term indicates the Kullback–Leibler divergence loss, and the second term indicates the reconstruction loss. $p(z|x)$ and $p(x|z)$ are probabilities to describe encoding and decoding processes, respectively.

For network learning module, we define the loss function by using cross entropy as:

$$\mathcal{L}_{NL} = -\sum_{v_i \in \mathcal{Y}_L} \sum_{u=1}^{U} Y_{iu} \ln H_{iu}, \tag{14}$$

where $\mathcal{Y}_L$ is the set of node indices that have labels, Y is the label indicator matrix, and U represents the dimension of the output embedding, which is equal to the number of categories.

The final loss of our AS-GCN is the linear combination of these two parts of loss with hyper-parameter λ to balance their weights, that is,

$$\mathcal{L} = \mathcal{L}_{NL} + \lambda \mathcal{L}_{NTM}. \tag{15}$$

E. Implementation

Our model is quite flexible and it is quite easy to incorporate some tricks when implementing our method. The tricks include, for example, supporting the use of the node level attention and the edge-type level attention, which are often used in the existing GNNs [27], [28].

First, considering the edge-type based neighbors of each node contribute to the embedding of the target node in different degrees, we adopt node level attention [7] to learn the importance of edge-type based neighbors for each node. To be specific, given a node pair (v_i, v_j) and a specified edge-type ϕ (where $\phi \in \{D, DT, TM, M\}$), the importance coefficient between nodes v_i and v_j can then be defined as:

$$b_{ij}^\phi = \text{LeakyReLU}(\eta_\phi^T [Wh_i || Wh_j]), \tag{16}$$

$$\alpha_{ij}^\phi = \text{softmax}_j(b_{ij}^\phi) = \frac{\exp(b_{ij}^\phi)}{\sum_{r \in N_i^\phi} \exp(b_{ir}^\phi)}, \tag{17}$$

where η_ϕ is the parameterized attention vector for edge-type ϕ, and W is the mapping matrix applied to each node. Then, the embedding of node v_i for edge-type ϕ can be aggregated by the neighbor's embeddings with its corresponding weight coefficients as:

$$h_i^\phi = \sigma(\sum_{j \in N_i^\phi} \alpha_{ij}^\phi W h_j). \tag{18}$$

Second, to best utilize the information from different types of edges, and learn the relation and the interaction between them, we adopt an edge-type level attention to fuse multiple semantic structure information which can be revealed by edge-type. Take a document node v_i, which mainly fuses information from edge-types D and DT, as an example, let h_i^ϕ denote its embedding under edge-type ϕ, the attention value β_i^ϕ can be represented as:

$$\beta_i^\phi = q^T \cdot \tanh(W_\phi \cdot (h_i^\phi)^T + b_\phi), \tag{19}$$

where q is the parameterized attention vector, W_ϕ is the weight matrix and b_ϕ is the bias vector.

After obtaining the attention value of each edge-type, i.e., β_i^D and β_i^{DT}, we normalize them via softmax function:

$$\gamma_i^\phi = \text{softmax}(\beta_i^\phi) = \frac{\exp(\beta_i^\phi)}{\sum_\phi \exp(\beta_i^\phi)}. \tag{20}$$

With the learned weights as coefficients, the l-th layer embedding of document node v_i can then be obtained by:

$$h_i^{(l)} = \Big\|_{\phi \in \{D, DT\}} \gamma_i^\phi \cdot h_i^{(\phi,l)}. \tag{21}$$

IV. EXPERIMENTS

We first give the experimental setup, and then compare our AS-GCN with state-of-the-arts on two network analysis tasks, i.e., node classification and network visualization. Next, we introduce the application on e-commerce search from JD. We finally investigate the hyper-parameter sensitivity.

A. Experimental Settings

Datasets. We adopt four publicly available datasets, as shown in Table II, to evaluate the performance of different methods.

TABLE II: Datasets descriptions.

Datasets	#Nodes	#Edges	#Categories
Hep-Small	397	812	3
Cora-Enrich	2,708	5,429	7
DBLP-Five	6,936	12,353	5
Hep-Large	11,752	134,956	4

- **Cora-Enrich**[1] is the text-rich version of the well-known citation network Cora dataset, where nodes are documents and edges are citation links. The textual description is collected from the titles, abstracts and all sentences from a document containing citations. Each paper is manually labeled as one of seven categories (*Case Based, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, and Theory*) based on their academic topics.
- **DBLP-Five** [13] includes a collection of documents in computer field, where the title and abstract are extracted as text for each document and the citation relationships are used to form links between documents. All the documents are divided into five categories (*High-Performance Computing, Software engineering, Computer networks, Theoretical computer science, and Computer graphics: Multimedia*) according to CCF (China Computer Federation) classification.
- **Hep-Small**[2] **and Hep-Large**[2] are two citation datasets about scientific documents in physics, where each node is associated with textual description composed of its title and abstract. Hep-Small contains 397 documents in three categories (*Nucl.Phys.Proc.Suppl, Phys.Rev.Lett, and Commun.Math.Phys*) connected by 812 links, and Hep-Large contains 11,752 documents in four categories (*Phys.Rev, Phys.Lett, Nucl.Phys, and JHEP*) connected by 134,956 links.

Baselines. We compare our proposed AS-GCN with the following baselines, including seven state-of-the-art methods and one variant of AS-GCN:

- **GCN** [6] is a semi-supervised graph neural network model which derives node representation by aggregating information from neighborhoods. Specifically, GCN is the base of our approach AS-GCN.
- **GAT** [7] introduces the masked self-attention mechanism to assign different neighbors with different specified weights.
- **DGI** [29] is an unsupervised graph neural network model which learns node embeddings by leveraging local mutual information maximization.
- **GraphSage** [30] is an inductive network embedding method which generalizes the aggregation beyond averaging to generate node emebddings for previously unobserved nodes.
- **AM-GCN** [31] is a GCN-based method which performs graph convolution over both topology and feature spaces.
- **Geom-GCN** [32] is a semi-supervised graph neural network model utilizing a geometric aggregation mechanism to obtain node representation.
- **BiTe-GCN** [17] is a semi-supervised graph neural network architecture through bidirectional convolution of topology and attributes.
- **AS-GCN-Two-Stage** is a variant of our AS-GCN. It removes the distribution sharing mechanism, and constructs a tri-typed network utilizing the fixed topic distribution and word distribution to perform convolution throughout the training.

Parameter settings. For all baselines, we use the source codes provided by the authors, and carefully turn parameters to get optimal performance. For our model, we utilize two-layer GCN as backbone, and employ pre-trained 300-dimensional GloVe embeddings [33] to initialize word embeddings. In each iteration of our framework, we select the top two topic nodes for each document node to set edges, and the top ten word nodes for each topic node to set edges. We set ReLU as the activation function and apply a dropout rate of 0.5 to further prevent overfitting. In addition, we set topic dimension in NTM to 100, weight decay to 5e-4, and λ to 0.8 to balance the loss of NTM and network learning module. We perform early stopping when validation loss does not decrease for 10 consecutive epochs. For all methods, we run 5 times with the same partition and report the average results. We use accuracy and macro F1-score to evaluate performance of models.

Training strategy. We consider some empirical training strategies similar as [34] to make our AS-GCN efficiently converge. Specifically, we pre-train NTM for 200 epochs employing an Adam optimizer with the learning rate of 1e-3, considering its convergence speed is much slower than that of GCN. In joint training, the NTM is trained with the learning rate of 5e-4, while the learning rate of network learning module is set to 5e-3 because the NTM is relatively stable.

B. Comparison with Baselines

We first make a quantitative comparison on node classification, and then a qualitative comparison on visualization. The results of node classification are shown in Table III.

Comparison with state-of-the-art methods. Based on the results, we make the following observations:

- Compared with all baselines, the proposed method AS-GCN consistently performs the best across different datasets. In particular, for ACC, AS-GCN achieves up to 20.51% on Hep-Small and 12.99% on Cora-Enrich relative improvements, respectively. These results illustrate the effectiveness of our AS-GCN.
- The overwhelming performance superiority of AS-GCN over backbone GCN implies that AS-GCN is capable of fully utilizing the most informative semantic structure information

[1] http://zhang18f.myweb.cs.uwindsor.ca/datasets/

[2] https://www.cs.cornell.edu/projects/kddcup/datasets.html

TABLE III: Comparisons on node classification.

Method	Hep-Small		Cora-Enrich		DBLP-Five		Hep-Large	
	Accuracy	**F1-score**	**Accuracy**	**F1-score**	**Accuracy**	**F1-score**	**Accuracy**	**F1-score**
GCN	0.6154	0.6183	0.8777	0.8596	0.9221	0.9142	0.4860	0.4862
GAT	0.6410	0.6334	0.8519	0.8454	0.9293	0.9234	0.4962	0.4917
DGI	0.5467	0.5468	0.7923	0.7694	0.8351	0.8080	0.4387	0.4197
GraphSage	0.5128	0.5134	0.8556	0.8481	0.9351	0.9068	0.4562	0.4256
AM-GCN	0.5897	0.5780	0.8852	0.8627	0.9163	0.9111	0.4409	0.4352
Geom-GCN	0.6410	0.6374	0.8963	0.8770	0.9307	0.9264	0.4834	0.4723
BiTe-GCN	0.6667	0.6749	0.9000	0.8936	0.9380	0.9368	0.5174	0.5058
AS-GCN-Two-Stage	0.6923	0.6939	0.9000	0.8921	0.9365	0.9290	0.5106	0.5042
AS-GCN	**0.7179**	**0.7128**	**0.9222**	**0.9107**	**0.9495**	**0.9454**	**0.5362**	**0.5341**

of text, so that the semantic architecture construction and the discriminative convolution reinforce each other.

- In comparison with BiTe-GCN designed specifically for text-rich network representation, our performance improvement further demonstrates the effectiveness of our new mechanism for designing an adaptive semantic architecture informed by the given learning objective.

Comparison with variants of AS-GCN. As shown in Table III, compared to GCN, AS-GCN-Two-Stage, which performs information propagation on an augmented tri-typed network, exhibits consistent improvement on all datasets. This validates the effectiveness of constructing a new architecture suitable for text-rich networks by utilizing the semantic structure information of text, including the local word sequence semantics and the global topic semantics. Furthermore, by introducing the distribution sharing mechanism, the results of AS-GCN are generally better than AS-GCN-Two-Stage on all datasets. This further confirms the necessity of designing an adaptive semantic architecture which learns both NTM and network learning module in an end-to-end manner to let them mutually enhance each other.

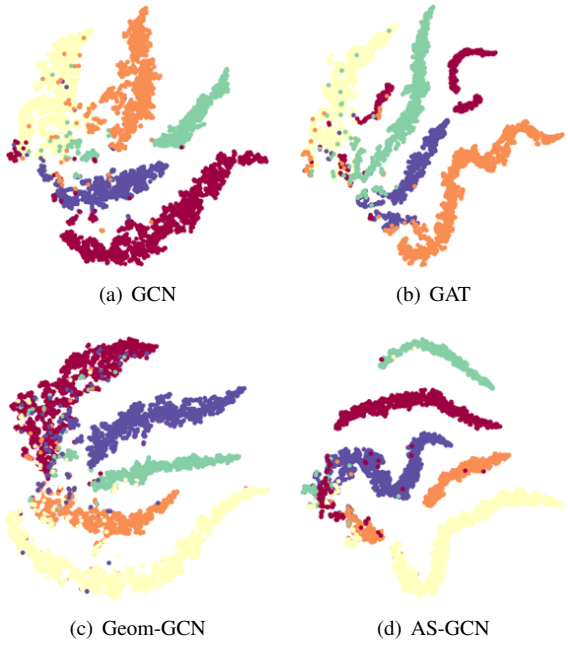
Visualization. For an intuitive comparison, we visualize the embeddings of some representative methods (i.e., GCN, GAT, Geom-GCN and our AS-GCN) on the DBLP dataset as an example. We utilize the well-known t-SNE tool [35] to project node embeddings to two dimensions. Different colors correspond to different categorical labels of documents.

As shown in Figure 3, GCN, GAT and Geom-GCN (which ignore the local word sequence semantics and the global topic semantics of texts) are less satisfactory, i.e., the documents belong to different categories are sometimes mixed with each other. The visualization of our AS-GCN performs best, where the learned embedding has a denser cluster structure, the highest intra-class similarity and the most distinctive boundaries among different classes.

C. Parameter Sensitivity

We demonstrate parameter sensitivity of AS-GCN using the DBLP dataset in Figure 4.

Analysis of topic embedding dimension K. We test the effect of the dimension of topic embedding, and vary it from 50 to 300. The result is shown in Figure 4(a). With the increase of the dimension of topic embedding, the accuracies increase first and then start to decrease. This parameter is relatively sensitive because the value of topic embedding dimension can directly influence the effectiveness of text information propagation.

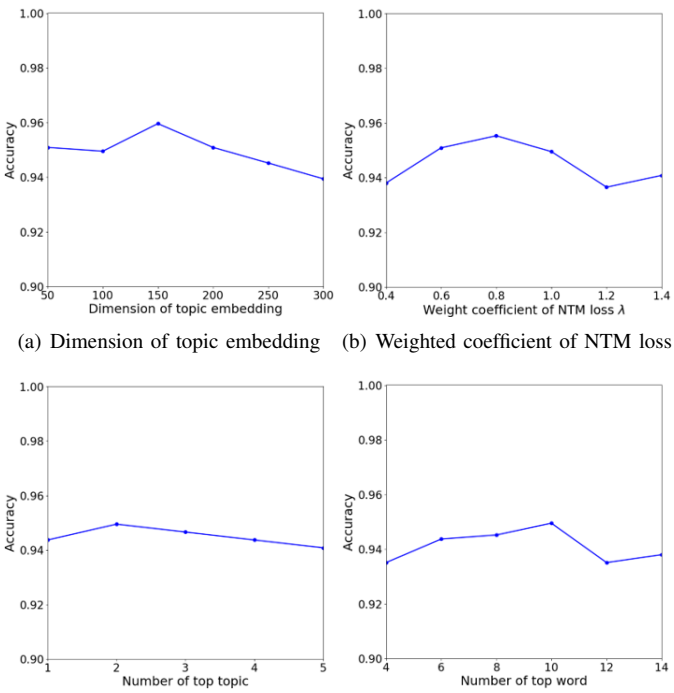


(a) GCN (b) GAT (c) Geom-GCN (d) AS-GCN

Fig. 3: The visualization of the node embeddings learned by (a) GCN, (b) GAT, (c) Geom-GCN and (d) AS-GCN on the DBLP dataset. Different colors correspond to different categorical labels in ground truth.

Analysis of weighted coefficient of NTM loss λ. To achieve the best performance of the model, we test the effect of the coefficient λ of NTM loss. The result is shown in Figure 4(b). With the increase of the coefficient λ, the performance shows a trend of first rising and then decreasing. It is reasonable since a too small coefficient of NTM loss would weaken the role of NTM on the process of network construction, whereas a too large coefficient would weaken the role of classification loss.

Analysis of the number of top topics. In order to check the impact of the number of edges between document nodes and topic nodes, we study the performance of AS-GCN with various number of edges ranging from 1 to 5 in Figure 4(c). For DBLP, the accuracies show a trend of first rising and then



(a) Dimension of topic embedding (b) Weighted coefficient of NTM loss

(c) Number of top topic (d) Number of top word

Fig. 4: Impact of hyper-parameters on DBLP.

slowly decreasing. This is also consistent with the fact that most documents in this dataset contain one or two topics [36].

Analysis of the number of top words. As for the number of edges between topic nodes and word nodes, we can see that with the growth of the numbers of edges, the performance also rises first, but the performance will drop quickly if the number of edges is larger than 10 for DBLP in Figure 4(d). It is probably because a small number of edges between topic nodes and word nodes would result in information loss and ineffective information propagation, whereas too many edges would introduce more noise.

V. Application on E-commerce Search

Dataset description. To further validate AS-GCN's effectiveness, we collect a JD e-commerce dataset and apply AS-GCN on this dataset to solve a classical e-commerce search problem from JD.com, i.e., the relevance estimation between the query and the item. The dataset contain 6.5M queries and 50M items. We assume that the topic information exists in these queries and items, and it is usually represented by the tree-structure category knowledge. For example, an item of 'red dress' belongs to 'clothing (first-level category) - women's wear (second-level category) - dress (third-level category)'. There are in total 191 first-level categories, 2,064 second-level categories and 19,570 third-level categories in our collected data. In addition, we extract 60K product phrases and 12K attribute phrases from this data, and use these phrases in AS-GCN model to achieve more accurate relevance estimation.

Baselines and metrics. We compare two kinds of baseline algorithms including text matching models and GCN models. Specifically, in text matching models, we compare two representation-based models (MV-LSTM [37] and ARC-I [38]) and three interaction-based models (K-NRM [39], ARC-II [38] and MatchPyramid [40]). The former models learn two separate embeddings of the query and item through a double-tower neural network structure. The latter models fuse the query's and item's embedding and input them into the unique neural network. DUET [41], as the mixed model of representation-based style and interaction-based style, is also used as one of our baseline algorithms. For GCN models, we compare AS-GCN with some representative models including GCN, GAT and BiTe-GCN which have been introduced in Section IV-A.

We choose six kinds of evaluation metrics to measure the model's quality. They are AUC (Area Under the receiver operating characteristic Curve), accuracy, precision, recall, F1-score and FNR (False Negative Rate). The lower FNR value implies the better model, while the other metrics are opposite.

Experimental results and analysis. The comparison results are shown in Table IV. Overall, GCN methods generally outperform text matching methods. For example, GCN outperforms MV-LSTM by 1.8% on AUC and 39.5% on FNR respectively. This improvement is mainly attributed to the positive aggregation function from the nodes' neighbor information. In GCN methods, BiTe-GCN has better results than GCN and GAT because of the distinctive mechanism of the bidirectional convolution of topology and attributes. However, BiTe-GCN does not consider the effect of different level's category information to the relevance estimation. AS-GCN ingeniously incorporates the category information as the topic information in order to better guide the relevance estimation, so that it outperforms all of the GCN methods including BiTe-GCN. Because the rate of the negative examples are higher than that of the positive examples in the search data, under the function of the neighbor information aggregation, it is easier to generate false negative examples for GCN methods than text matching methods and this is why GCN methods under-perform text matching methods in Recall.

VI. Related Work

In line with the focus of our work, we first briefly review the most related work on graph neural networks (GNNs) for text-rich network representation and text analysis, and then introduce methods for solving the homophily assumption of GNNs.

A. GNNs for Text-Rich Network Representation

Several recent works have studied the text-rich network representation. For example, Chen *et al*. [42] augment the complex e-commercial data into a text-rich heterogeneous e-commercial network to produce meaningful representations for applications such as produce classification. Shi *et al*. [16] propose an unsupervised framework HyperMine, which exploits multi-granular contexts and combines signals from both text and network, to discover hypernymy in text-rich networks. Shang

TABLE IV: Comparisons on e-commerce dataset. The metric of AUC is the most important evaluation metric.

Types	Methods	AUC(*)	Accuracy	Precision	Recall	F1-score	FNR
Text matching methods	MV-LSTM	0.8278	0.8023	0.8021	0.9873	0.8851	0.8224
	K-NRM	0.8021	0.7918	0.7942	0.9854	0.8796	0.8618
	ARC-I	0.7345	0.7771	0.7769	0.9975	0.8735	0.9669
	ARC-II	0.7783	0.7920	0.7915	0.9915	0.8803	0.8815
	MatchPyramid	0.8007	0.7946	0.7988	0.9806	0.8805	0.8336
	DUET	0.8077	0.7799	0.7789	**0.9980**	0.8749	0.9563
GCN methods	GCN	0.8458	0.8580	0.8816	0.9425	0.9110	0.4272
	GAT	0.8523	0.8539	0.8819	0.9361	0.9082	0.4234
	BiTe-GCN	0.8564	0.8598	0.8847	0.9409	0.9119	0.4139
	AS-GCN	**0.8614**	**0.8600**	**0.8850**	0.9408	**0.9120**	**0.4128**

et al. [43] present a hierarchical embedding and clustering framework which consumes a text-rich network as the input for automatic topic taxonomy construction. Very recently, Wang *et al*. [44] present a community-enhanced retrieval model for text-rich heterogeneous information networks to improve retrieval accuracy (content relevance). Jin *et al*. [17] introduce a new graph convolutional network architecture via bidirectional convolution of topology and attributes on text-rich networks.

B. GNNs for Text Analysis

As GNNs become the dominant tools for network representation learning, several efforts have been made to apply GNNs for boosting performance of text analysis. For example, Text GCN [45] constructs a heterogeneous word document network for a corpus based on word co-occurrence and document word relations, and turns document classification into a node classification problem. TensorGCN [26] utilizes the semantic, syntactic, and sequential contextual information from text to construct a network, and then builds a network-based learning framework which performs intra-graph and inter-graph propagation to realize text classification. HeteGCN [46] simplifies Text GCN by dissecting into several HeteGCN models, so as to learn feature embeddings and derive document embeddings.

C. Homophily Assumption

In recent years, the limitation of homophily assumption of GNNs has drawn considerable attention [15]. Existing methods for relieving the homophily assumption can be generally divided into four families, that is, topology optimization, self-supervised, skip connection and attention-based methods.

The first family usually adopts the idea of topology optimization to improve GNNs. DropEdge [3] proposes to reduce the message passing by randomly deleting a certain number of edges from the input network. Geom-GCN [32] proposes a novel geometric aggregation scheme to overcome neighborhood structural information loss and the lack of long-range dependencies. The second family typically augments the original label set by adding the high-credible labels derived from GNNs. M3S [47] proposes a multi-stage training algorithm which first adds confident data with virtual labels to the label set, and then applies DeepCluster on the embedding process of GNNs. The third family adaptively selects the appropriate neighborhoods for each node from the perspective of jumping knowledge. JKNet [48] introduces jumping knowledge networks, which flexibly leverages different neighborhood ranges for each node, to enable better structure-aware representation. In addition, some attention-based works can also be considered to solve the topological limitations of GNNs. GAT [7] introduces the attention mechanisms to allocate different neighbors with different weights.

Though those methods improve the performance of GNNs on learning representations of text-rich networks, they still have several limitations. That is, they fail to fully embody the most informative semantic structure information of text in the process of modeling. At the same time, the network structure we utilized is often imperfect. Therefore, it is of great significance to explore an adaptive architecture informed by the ground truth.

VII. CONCLUSION

We propose a new adaptive semantic architecture of graph neural network, namely AS-GCN, which unifies neural topic model and GCN, for text-rich network representation. By integrating neural topic model and GCN in a unified framework, our model can embed richer structural semantics, including the local word sequence structure and the global topic structure, in the learned representation to make the model more powerful. Meanwhile, by introducing the network learning module which performs a semantic-aware propagation of information on the augmented tri-typed network, our method can not only effectively alleviate the homophily assumption of the previous GCN methods, but also well realize the optimal balance between topology and attributes, that is, learning the contributions of network part and text part automatically aiming to the given learning objectives such as node classification.

Experimental results on several text-rich networks demonstrate that our new adaptive semantic architecture has a significant improvement over the state-of-the-art methods. Moreover, this architecture is well applied in e-commerce search scenes from JD. Last but not the least, this architecture is almost orthogonal to most existing GNNs and thus can be readily incorporated into various GNNs to further improve their performance.

VIII. Acknowledgements

This work is supported in by the Natural Science Foundation of China under grants 61772361, 61876128 and 62172052.

References

[1] Q. Long, Y. Jin, Y. Wu, and G. Song, "Theoretically improving graph neural networks via anonymous walk graph kernels," in *Proceedings of the WWW*, pp. 1204–1214, 2021.

[2] D. Jin, C. Huo, C. Liang, and L. Yang, "Heterogeneous graph neural network via attribute completion," in *Proceedings of WWW*, pp. 391–400, 2021.

[3] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *Proceedings of ICLR*, 2020.

[4] J. You, R. Ying, and J. Leskovec, "Position-aware graph neural networks," in *Proceedings of ICML*, pp. 7134–7143, 2019.

[5] Z. Liu, M. Wan, S. Guo, K. Achan, and P. S. Yu, "Basconv: Aggregating heterogeneous interactions for basket recommendation with graph convolutional neural network," in *Proceedings of SDM*, pp. 64–72, 2020.

[6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *proceedings of ICLR*, 2017.

[7] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proceedings of ICLR*, 2018.

[8] G. Bachmann, G. Bécigneul, and O. Ganea, "Constant curvature graph convolutional networks," in *Proceedings of ICML*, pp. 486–496, 2020.

[9] J. Chen, T. Ma, and C. Xiao, "Fastgcn: Fast learning with graph convolutional networks via importance sampling," in *Proceedings of ICLR*, 2018.

[10] D. Jin, J. Huang, P. Jiao, L. Yang, D. He, F. Fogelman-Soulié, and Y. Huang, "A novel generative topic embedding model by introducing network communities," in *Proceedings of WWW*, pp. 2886–2892, 2019.

[11] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proceedings of IJCAI*, pp. 2111–2117, 2015.

[12] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A fair comparison of graph neural networks for graph classification," in *Proceedings of ICLR*, 2020.

[13] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of SIGKDD*, pp. 990–998, 2008.

[14] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," in *Proceedings of NeurIPS*, 2020.

[15] J. Zhu, R. A. Rossi, A. B. Rao, T. Mai, N. Lipka, N. K. Ahmed, and D. Koutra, "Graph neural networks with heterophily," in *Proceedings of AAAI*, 2021.

[16] Y. Shi, J. Shen, Y. Li, N. Zhang, X. He, Z. Lou, Q. Zhu, M. Walker, M. Kim, and J. Han, "Discovering hypernymy in text-rich heterogeneous information network by exploiting context granularity," in *Proceedings of CIKM*, pp. 599–608, 2019.

[17] D. Jin, X. Song, Z. Yu, Z. Liu, H. Zhang, Z. Cheng, and J. Han, "Bite-gcn: A new GCN architecture via bidirectional convolution of topology and features on text-rich networks," in *Proceedings of WSDM*, pp. 157–165, 2021.

[18] D. Jin, Z. Yu, P. Jiao, S. Pan, P. S. Yu, and W. Zhang, "A survey of community detection approaches: From statistical modeling to deep learning," *TKDE*, 2021.

[19] F. Hu, Y. Zhu, S. Wu, L. Wang, and T. Tan, "Hierarchical graph convolutional networks for semi-supervised node classification," in *Proceedings of IJCAI*, pp. 4532–4539, 2019.

[20] L. Yang, Z. Kang, X. Cao, D. Jin, B. Yang, and Y. Guo, "Topology optimization based graph convolutional network," in *Proceedings of IJCAI*, pp. 4054–4061, 2019.

[21] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," in *Proceedings of AAAI*, pp. 3950–3957, 2021.

[22] Y. Miao, E. Grefenstette, and P. Blunsom, "Discovering discrete latent topics with neural variational inference," in *Proceedings of ICML*, vol. 70, pp. 2410–2419, 2017.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of ICLR*, 2014.

[24] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439–453, 2020.

[25] I. Sato and H. Nakagawa, "Stochastic divergence minimization for online collapsed variational bayes zero inference of latent dirichlet allocation," in *Proceedings of SIGKDD*, pp. 1035–1044, 2015.

[26] X. Liu, X. You, X. Zhang, J. Wu, and P. Lv, "Tensor graph convolutional networks for text classification," in *Proceedings of AAAI*, pp. 8409–8416, 2020.

[27] Y. Yang, X. Wang, M. Song, J. Yuan, and D. Tao, "SPAGAN: shortest path graph attention network," in *Proceedings of IJCAI*, pp. 4099–4105, 2019.

[28] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proceedings of WWW*, pp. 2022–2032, 2019.

[29] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proceedings of ICLR*, 2019.

[30] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of NeurIPS*, pp. 1024–1034, 2017.

[31] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, "AM-GCN: adaptive multi-channel graph convolutional networks," in *Proceedings of KDD*, pp. 1243–1253, 2020.

[32] H. Pei, B. Wei, K. C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," in *Proceedings of ICLR*, 2020.

[33] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of EMNLP*, pp. 1532–1543, 2014.

[34] P. Cui, Y. Liu, and B. Liu, "A neural topic model based on variational auto-encoder for aspect extraction from opinion texts," in *Proceedings of NLPCC*, vol. 11838, pp. 660–671, 2019.

[35] V. D. M. Laurens and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, 2008.

[36] D. Jin, K. Wang, G. Zhang, P. Jiao, D. He, F. Fogelman-Soulié, and X. Huang, "Detecting communities with multiplex semantics by distinguishing background, general, and specialized topics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2144–2158, 2020.

[37] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations.," in *Proceedings of AAAI*, pp. 2835–2841, 2016.

[38] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proceedings of NeurIPS*, pp. 2042–2050, 2014.

[39] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proceedings of SIGIR*, pp. 55–64, 2017.

[40] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition.," in *Proceedings of AAAI*, vol. 16, pp. 2793–2799, 2016.

[41] B. Mitra, F. Diaz, and N. Craswell, "Learning to match using local and distributed representations of text for web search," in *Proceedings of WWW*, pp. 1291–1299, 2017.

[42] W. Chen, C. Liu, J. Yin, H. Yan, and Y. Zhang, "Mining e-commercial data: A text-rich heterogeneous network embedding approach," in *Proceedings of IJCNN*, pp. 1403–1410, 2017.

[43] J. Shang, X. Zhang, L. Liu, S. Li, and J. Han, "Nettaxo: Automated topic taxonomy construction from text-rich network," in *Proceedings of WWW*, pp. 1908–1919, 2020.

[44] L. Wang, X. Yu, and F. Tao, "A community-enhanced retrieval model for text-rich heterogeneous information networks," in *Proceedings of ICDM Workshops*, pp. 505–513, 2019.

[45] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of AAAI*, pp. 7370–7377, 2019.

[46] R. Ragesh, S. Sellamanickam, A. Iyer, R. Bairi, and V. Lingam, "Hetegcn: Heterogeneous graph convolutional networks for text classification," in *Proceedings of WSDM*, pp. 860–868, 2021.

[47] K. Sun, Z. Lin, and Z. Zhu, "Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes," in *Proceedings of AAAI*, pp. 5892–5899, 2020.

[48] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proceedings of ICML*, vol. 80, pp. 5449–5458, 2018.