TeKo: Text-Rich Graph Neural Networks With External Knowledge

Zhizhi Yu, Di Jin, Member, IEEE, Jianguo Wei, Member, IEEE, Yawen Li, Ziyang Liu, Yue Shang, Jiawei Han, Fellow, IEEE, and Lingfei Wu, Member, IEEE

Abstract—Graph neural networks (GNNs) have gained great prevalence in tackling various analytical tasks on graph-structured data (i.e., networks). Typical GNNs and their variants adopt a message-passing principle that obtains network representations by the attribute propagates along network topology, which however ignores the rich textual semantics (e.g., local word-sequence) that exist in numerous real-world networks. Existing methods for text-rich networks integrate textual semantics by mainly using internal information such as topics or phrases/words, which often suffer from an inability to comprehensively mine the textual semantics, limiting the reciprocal guidance between network structure and textual semantics. To address these problems, we present a novel text-rich GNN with external knowledge (TeKo), in order to make full use of both structural and textual information within text-rich networks. Specifically, we first present a flexible heterogeneous semantic network that integrates high-quality entities as well as interactions among documents and entities. We then introduce two types of external knowledge, that is, structured triplets and unstructured entity descriptions, to gain a deeper insight into textual semantics. Furthermore, we devise a reciprocal convolitional mechanism for the constructed heterogeneous semantic network, enabling network structure and textual semantics to collaboratively enhance each other and learn high-level network representations. Extensive experiments illustrate that TeKo achieves state-of-the-art performance on a variety of text-rich networks as well as a large-scale e-commerce searching dataset.

Index Terms—External knowledge, graph neural networks (GNNs), network representation, text-rich networks.

I. INTRODUCTION

NETWORKS are ubiquitous structures for abstracting and modeling things and their relationships, such as citation networks, bibliographic networks, as well as biomedical networks. With the popularity of deep learning, network representation has become a hot research topic and widely applied in different downstream tasks. Lately, graph neural networks (GNNs) [1], [2], which exhibits significant power on naturally capturing both network structures and attribute information, have gained great success and have been adapted to a wide range of application tasks, including community detection [3] and recommender system [4].

The classical GNNs and their variants [5], [6] adopt a message-passing principle, in which the most essential part is the attribute propagates along network topology. However, networks in reality usually consist of nodes with rich textual description, which are called text-rich networks. Typical examples include academic networks (e.g., Cora) in which document nodes are accompanied by their abstracts, as well as e-commerce networks (e.g., Amazon) where product nodes are attached with their descriptions. Under text-rich situation, existing GNNs may suffer from poor performance, since the propagation mechanism within node neighborhoods typically...
only treat textual information as attribute words, as shown in Fig. 1(a), inevitably leading to the loss of some important semantic structure information (e.g., local word-sequence or global topic) contained in the text.

A few very recent studies [7], [8] have been dedicated to generalizing GNNs to text-rich networks. They incorporate semantic structures within the textual information, including local word-sequence and/or global topics, to the original network structure for modeling text-rich networks. While doing so can leverage additional semantic information in texts to a certain extent, these methods fail to well comprehend the semantic content of network data space and reason over complex concepts and relational paths. For example, given two entities identified from corpus, “mobile phone” and “mobile phone case,” if we only model the semantic structure in network data space, it may make the representation of these two entities very similar, negatively influencing the performance of GNNs. Our intuition is that, leveraging the knowledge (e.g., the concept of “mobile phone” and “mobile phone case”) provided from outside sources (e.g., Wikipedia [9] and ConceptNet [10]) to gain a deep insight into textual semantics and further facilitate the prediction of downstream tasks.

So, an intriguing yet important question is how to effectively leverage external knowledge to design text-rich GNNs. There are two key challenges to consider. First, how to identify appropriate and useful related knowledge in order to well comprehend textual semantics underlying text-rich networks? External knowledge usually constitutes multitype data. For example, Wikipedia contains two-type data, that is, structured triplets as well as unstructured entity descriptions. Different data types represent different semantics, each of which reflects one aspect of textual information [11], [12]. Therefore, it is important to comprehend the textual semantics via the reciprocal fusion of different data types of external knowledge. Second, how to fully understand and leverage the acquired knowledge to facilitate the effective guidance of knowledge space to text-rich network data space? As the structure and information contained in knowledge space and text-rich network data space are different, naively gluing information from these two spaces together may lead to an over-sophisticated model [13]. As a result, it is extremely significant to design a more advanced approach that can flexibly consider not only the information aggregation of each space but also the information interaction among different spaces. More importantly, the model itself should also correctly and adaptively learn the contribution of balance of network structure and textual information within text-rich networks aiming at given learning objectives.

In this article, we pay close attention to the above problems of text-rich network representations and present a new Text-rich GNN with external Knowledge, namely TeKo. To this end, we first augment the text-rich network with a newly constructed heterogeneous document-entity network, as presented in Figs. 1(b) and 2, so as to incorporate entities and capture rich semantic structures among documents and entities. We then introduce a knowledge-based entity representation module to adaptively extract useful external knowledge from structured triplets and unstructured entity descriptions. In this way, the external knowledge is capable of helping comprehend the semantic content of network data space. Finally, we design a discriminative propagation mechanism, based on reciprocal graph attention, to realize the interaction between both network structure and textual semantics. During the intermediate training steps, these two parts are guided by each other and optimized collaboratively. Meanwhile, by making full use of textual semantics, TeKo can also alleviate the topological limitations of GNNs [14] such as heterophily.

The main contributions are summarized as follows.

1) We gain a deep insight into the textual semantics underlying text-rich networks from the perspective of knowledge enhancement, empowering the effective integration of both structural and textual information.

2) We present a new GNN, namely TeKo, which innovatively employs the guidance of external knowledge space over network data space, for text-rich network representations.

3) Extensive experiments illustrate that the proposed TeKo significantly outperforms the state-of-the-arts across various real-world text-rich networks as well as a large-scale e-commerce searching dataset.

II. Preliminaries
We first present the terms and notations, and then give the problem definitions. We finally discuss GNNs which serve as the base of our proposed TeKo.

A. Terms and Notations
Given an undirected and unweighted text-rich network $G = (R, V, E)$, where $R$ is a set of document raw text, $V = \{v_1, \ldots, v_n\}$ and $E = \{e_{ij}\} \subseteq V \times V$ are the sets of nodes and edges, respectively. The topological structure of $G$ is represented by an adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, where $a_{ij} = 1$ if nodes $v_i$ and $v_j$ are connected, or $a_{ij} = 0$ otherwise. The attribute matrix of $G$ is denoted as $X \in \mathbb{R}^{n \times f}$, in which attributes are generated from document raw text $R$ and $f$ represents the attribute dimension. Frequently used notations are displayed in Table I.

| Notations | Explanations |
|-----------|--------------|
| $G$       | A text-rich network. |
| $R$       | The set of document raw text. |
| $V, E$    | The sets of nodes and edges of a network. |
| $e_{ij}$  | The edge between nodes $v_i$ and $v_j$. |
| $a_{ij}$  | The connection between nodes $v_i$ and $v_j$. |
| $A, X$    | The adjacency matrix and node attribute matrix. |
| $D$       | The node degree matrix. |
| $x_i$     | The attribute vector of a given node $v_i$. |
| $\{h, r, t\}$ | A triplet in knowledge graph. |
| $h_i$     | The latent representation of a given node $v_i$. |
| $e_s$     | The triple representation of an entity node $w_i$. |
| $e_d$     | The textual representation of an entity node $w_i$. |
| $\sigma$  | The non-linear activation function. |
| $N_i$     | The set of local neighbors of node $v_i$. |
| $\alpha$  | Weight of type-level attention. |
| $\beta$   | Weight of node-level attention. |

B. Problem Definitions
We focus on two kinds of downstream tasks, namely, semi-supervised node classification and unsupervised node clustering, to assess the learned text-rich network representations.
where $h_i$ represents the layer-specific trainable transformation matrix, and $\sigma$ is the nonlinear activation function. GNNs work well in learning network representations and applying them to different analytical tasks [18], [19], but numerous real-world networks contain rich textual information. Since existing GNNs typically regard text as attribute words alone, they will inescapably overlook important textual semantics. Therefore, it is of great significance to design a new text-rich GNN that fully utilize both network structure and textual semantics.

### III. METHODOLOGY

We start with a brief overview of the proposed TeKo, and then introduce the details of three key components.

#### A. Overview

To let the textual semantics essentially provide supplementary information for text-rich network representation, we present a new text-rich GNN that can effectively integrate network structure and textual semantics via the effective guidance of external knowledge space over network data space, namely TeKo. The whole structure of the proposed approach is displayed in Fig. 3, which is constituted of three major parts: semantic network generation, knowledge-based entity representation, as well as heterogeneous graph attention. The semantic network generation is designed to comprehensively capture textual semantics at a structural level. It introduces the relationship of texts in the knowledge space, and accordingly augments the original text-rich network into a flexible heterogeneous document-entity network, modeling textual semantics from both local and global angles. The knowledge-based entity representation is designed to utilize external knowledge to analyze textual semantics from multiple perspectives, and further enhance the integration of textual semantics and network structures. It extracts useful and related knowledge from structured triples and unstructured entity descriptions, and learns jointly entity representations by automatically finding a balance between these two types of information. The heterogeneous graph attention is designed to realize aggregation of the same spatial information and interaction between different spatial information. It introduces a discriminative mechanism based on reciprocal graph attention, enabling the model to effectively balance network structure and textual semantics.

#### B. Semantic Network Generation

As an important auxiliary information of text-rich networks, textual semantics (e.g., local word-sequence) play an

---

**Fig. 2.** Illustrative example of a text-rich network incorporating entities, where entities are annotated by TagMe [15].

**Fig. 3.** The whole structure of the proposed approach: TeKo. The whole structure of the proposed approach is displayed in Fig. 3, which is constituted of three major parts: semantic network generation, knowledge-based entity representation, as well as heterogeneous graph attention. The semantic network generation is designed to comprehensively capture textual semantics at a structural level. It introduces the relationship of texts in the knowledge space, and accordingly augments the original text-rich network into a flexible heterogeneous document-entity network, modeling textual semantics from both local and global angles. The knowledge-based entity representation is designed to utilize external knowledge to analyze textual semantics from multiple perspectives, and further enhance the integration of textual semantics and network structures. It extracts useful and related knowledge from structured triples and unstructured entity descriptions, and learns jointly entity representations by automatically finding a balance between these two types of information. The heterogeneous graph attention is designed to realize aggregation of the same spatial information and interaction between different spatial information. It introduces a discriminative mechanism based on reciprocal graph attention, enabling the model to effectively balance network structure and textual semantics.
indispensable role in learning network representations. To take full advantage of the underlying semantic structures within node textual information, we construct a heterogeneous semantic network that integrates entities and corresponding semantic structures. The semantic network generation consists of two parts: entity network generation as well as whole network integration, as shown in Fig. 2.

1) Entity Network Generation: As motivated, on a text-rich network, it is imperative to capture the underlying textual semantics. However, it will inevitably introduce a lot of useless information if we treat all words contained in the corpus as entities. To this end, we consider recognizing high-quality entities from corpus and associating them with Wikipedia by means of entity linking tool TagMe [15], that is, select entities above a predefined threshold $\delta_{\text{tag}}$. After that, the edges among entities can be built according to the similarity of their initial knowledge-based representation (see Section III-C below). Actually, there are many ways to construct entity edges such as point mutual information or heat kernel. Here, we select the cosine similarity to generate edges between entities.

Cosine Similarity: It calculates the entity similarity utilizing the cosine value of angle among attribute vectors. Mathematically, given a text-rich network $G_D = (R, V_D, E_D)$, in which every document node $v_i$ has a unique textual information denoted as $d_i$. Let $V_W$ be the set of annotated entities, $e_i$ be the knowledge-based representation of entity node $w_i$, the similarity $s_{ij}$ between entity nodes $w_i$ and $w_j$ can be defined in the following as

$$s_{ij} = \frac{e_i \cdot e_j}{|e_i||e_j|}.$$  \hfill (3)

Then, the adjacency matrix $E_W$ can be obtained by choosing node pairs where the similarity value exceeds a certain threshold $\delta_{\text{sim}}$. Note that the constructed entity sub-network is static, where entity nodes and edges do not alter during training.

2) Whole Network Integration: We finally augment the original text-rich network into a heterogeneous semantic network, so as to explicitly capture both network topology and textual semantics. It includes two types of nodes (i.e., document nodes and entity nodes), and three types of edges (i.e., edges among document nodes from the original text-rich network representing paper relationships, edges between document nodes and entity nodes constructed based on the inclusion relationships between documents and entities, and edges among entity nodes capturing word-associated semantics). Formally, the heterogeneous semantic network is represented as

$$G = (V_D \cup V_W, E_D \cup E_{DW} \cup E_W)$$  \hfill (4)

where $E_{DW}$ is the set of edges between document nodes and entity nodes. In this way, by incorporating entities and relations, we can capture the abundant semantics within text-rich networks at a structure level.

3) Discussion: For semantic network generation, the most time-consuming part is the construction of an entity network. Supposing there are $|V_W|$ entities recognized by the entity linking tool TagMe from corpus, and the entity feature dimension is $u$, the calculation of the entity network using cosine similarity needs $O(|V_W|^2u)$ time. In practice, we would pick node pairs where the similarity value exceeds a certain threshold $\delta_{\text{sim}}$ to sparse the entity network. Then quick algorithms such as KD-tree [20] and Ball-tree [21] could be used to reduce its complexity to $O(\log_2|V_W|)$.
C. Knowledge-Based Entity Representation

To further comprehensively mine textual semantics underlying text-rich networks and facilitate their interaction with text-rich network structure, we learn the entity representation using external knowledge. Considering that the information density of external knowledge is usually sparse and incomplete, we generate entity representation by encoding both structured triplets and unstructured entity descriptions, and adaptively integrate them with a learnable gating mechanism.

1) Triplet Representation: Knowledge graph embedding is a powerful tool for learning representations of entities and their relations from structured triplets. Here, we employ TransE [22], a simple and efficacious approach, to parameterize triplets to learn entity representations $e_i \in \mathbb{R}^n$. Given a triplet $(h, r, t)$, let $r$ be the representations of relation $r$, $h$, and $t$ be the representations of entities $h$ and $t$, respectively. TransE aims to embed every entity and relation by optimizing the translation rule $h + r \approx t$, if $(h, r, t)$ holds

$$f(h, r, t) = -\|h + r - t\|_2^2$$

in which $h$ and $t$ are subject to the normalization constraint, that is, the magnitude of each vector is 1. Intuitively, a large score of $f(h, r, t)$ indicates that the triplet tends to be a real fact, and vice versa. Note that, we only consider triplets in which entity nodes within the heterogeneous semantic network are head (subject) instead of tail (object).

2) Textual Representation: For each entity, we employ the paragraph of its corresponding Wikipedia page as the description. As the description may represent an entity from various aspects, we adopt latent Dirichlet allocation (LDA) [23], an unsupervised text generation approach, to learn the textual representation of each entity (denoted as $e_d \in \mathbb{R}^n$). Specifically, LDA assumes that each description comes from a mixture of topics, where the topics are shared across the descriptions and the mixing proportion of each description is unique. The generation process for each description can be iteratively divided into two steps: the first is to choose a topic with a certain probability, and the second is to select a word under this topic with a certain probability.

3) Representation Fusion: Since both the structured triplets and unstructured description provide valuable information for an entity, we jointly integrate these two kinds of information for knowledge-based entity representation. To achieve the optimal combination of triple representation $e_t$ and textual representation $e_d$, we introduce a learnable gating mechanism [24] to determine how much the joint representation depends upon triplets or description. The gating mechanism is widely applied in many deep learning fields, such as Natural Language Processing (NLP) and Computer Vision (CV). It can be regarded as an information regulator which adaptively controls how much of the output is produced by representations from different sources. Mathematically, given an entity node $v_i$, the joint representation $e_i$ can be formulated as

$$e_i = g_e \odot e_t + (1 - g_e) \odot e_d$$

where $g_e \in \mathbb{R}^n$ is a gating vector with elements in $[0, 1]$ to balance the information from triplets and description, and $g_e$ represents element-wise multiplication. Obviously, the joint representation with gate closer to 0 tends to use textual representation; whereas the joint representation with gate closer to 1 utilizes triple representation. More importantly, to restrain the value of each element in the scope of 0 to 1, we apply sigmoid function to calculate the gate $g_e$.

$$g_e = \text{sigmoid}(\tilde{g}_e)$$

where $\tilde{g}_e$ is a real-value vector learned during training.

D. Heterogeneous Graph Attention

The core of our approach is to sufficiently utilize both network structure and textual semantics for text-rich network representations with external knowledge. For this purpose, we introduce a heterogeneous graph attention that performs information propagation on the augmented heterogeneous semantic network. It considers not only the information aggregation in the text-rich network data space but also the information guidance of the knowledge space to the network data space.

Owing to the heterogeneity of nodes in the augmented semantic network, different types of nodes may be located in different feature spaces. Therefore, for a node $v_i$ with type $\phi_i$, we project its features into common space using a type-specific transformation matrix $W_{\phi_i}$ as

$$h_i' = W_{\phi_i} \cdot h_i$$

where $h_i'$ is the projected feature of node $v_i$.

After that, to facilitate information propagation among neighboring nodes, we learn node representations from the perspective of network schema, which makes a heterogeneous network semi-structured, guiding the exploration of semantics. Particularly, given a target node $v_i$, neighbors with different types may have different effects on it. Therefore, we employ type-level attention [25] to estimate the significance of neighbors with different types. Let $h_j$ be the embedding of type $\phi_j$ which is defined as the sum of the neighbor embedding $h_j'$ with node $v_j \in \mathcal{N}_i$ under type $\phi_j$, that is

$$h_{\phi_i} = \sum_{v_j} \hat{a}_{ij} h_j'$$

where $\hat{A} = [\hat{a}_{ij}]$ is defined in Eq. (2).

Then, the type-level attention weights can be calculated by the target node embedding $h_i'$ and the type embedding $h_{\phi_i}$

$$\alpha_{\phi} = \text{softmax}_{\phi} (\sigma (\eta_{\phi}^T [h_i', h_{\phi_i}])))$$

where $\eta_{\phi}$ is the attention vector for the type $\phi$, $\sigma$ represents the activation function such as LeakyReLU, and the softmax function is adopted to normalize across all the types.

On the other hand, considering that different neighbors of the same type could also have different significances, we further apply node-level attention [26] to estimate the weights between nodes with the same type. Formally, given a target node $v_i$ with type $\phi_i$, let $v_j$ be its neighboring node with type $\phi_j$, the node-level attention weights can then be computed as

$$\beta_{ij} = \text{softmax}_{v_j} (\sigma (\gamma^T \cdot \alpha_{\phi_j} [h_i', h_{\phi_i}])))$$

where $\gamma$ is the attention vector, and softmax is applied to normalize across all the neighbors of target node $v_i$.  

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
By integrating the above process, the matrix form of the layer-wise propagation principle in heterogeneous graph attention can be written as

$$H^{(k)} = \sigma \left( \sum_{\phi \in \Phi} B_{\phi} \cdot H^{(k-1)}_{\phi} \cdot W^{(k-1)}_{\phi} \right)$$

where $\Phi$ is the set of node types in the augmented semantic network, and $B = [\beta_{ij}]$ represents the attention matrix. In this way, we can well realize reciprocal enhancement of information from both network structure and textual semantics with the guidance of external knowledge.

### E. Model Training

After obtaining the final document representations, we can apply them to various network analytical tasks and devise corresponding loss functions. For semi-supervised node classification, the loss function is defined by adopting cross entropy

$$L = - \sum_{i \in V_L} \sum_{c=1}^{C} y_i[c] \log h_i[c]$$

where $V_L$ denotes the node index set with labels, $C$ is the number of classes, and $y_i$ is the true one-hot label of node $v_i$. For unsupervised node clustering, without any node labels, the loss function is defined via negative sampling as

$$L = - \sum_{(v_i,v_j) \in \Omega} \log \sigma (h_i^T \cdot h_j) - \sum_{(v_m,v_n) \in \Omega^-} \log \sigma (-h_m^T \cdot h_n)$$

where $\sigma$ represents activation function, $\Omega$ is the set of positive node pairs, $\Omega^-$ is the set of negative node pairs (the complement of $\Omega$). The detailed process of TeKo is summarized in Algorithm 1.

#### Algorithm 1 TeKo Algorithm

**Input:** Text-rich network $G = (R, V_D, E_D)$, external knowledge Wikidata5M, epoch $T$, threshold $\delta_{arg}$ and $\delta_{sim}$;

**Output:** Node representation $H$;

1. Recognize high-quality entities $V_W$ from raw text $R$;
2. for each entity node $w_i$, do
3. Calculate triplet representation $e_s$;
4. Calculate textual representation $e_t$;
5. Obtain knowledge-based representation $e_i$ by Eq. (6);
6. end for
7. Calculate the similarity of entity nodes by Eq. (3) and obtain $E_W, E_{DW}$;
8. while $t < T$ do
9. Calculate $H$ using Eq. (8-12);
10. Calculate the loss function using Eq. (13) or Eq. (14);
11. Back propagation and update parameters;
12. end while
13. return $H$

### IV. EXPERIMENTS

We first introduce the experimental setup. We then evaluate the new approach TeKo on three network analysis tasks, thereafter introduce a deep investigation on different components of TeKo and give the parameter analysis. We finally present the e-commerce searching application.

| 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           |

A. Experimental Setup

1) **Datasets:** We conduct experiments on four real-world text-rich datasets, where the statistics are displayed in Table II.

2) **Baselines:** We evaluate the performance of our proposed TeKo by comparing it with ten state-of-art baselines.

1) **GCN** [16] is a classical GNN which derives node representations by defining convolutional operators on graph-structured data.

2) **GAT** [26] is an attention-based GNN which performs convolutional operations in the graph spatial domain and assigns different weights to neighbors.

3) **SGC** [28] is a simplified GNN that reduces the complexity of model by eliminating nonlinearities and weight matrices among neighboring layers.

4) **DGI** [29] is an unsupervised GNN maximizing local mutual information by patch representation of graph.

5) **GMI** [30] is an unsupervised GNN that measures the correlation between input graphs and high-level representations through graphical mutual information.

6) **GraphSage** [31] is an inductive GNN leveraging sampler and aggregator to generate node representations.

7) **AM-GCN** [32] is an adaptive multichannel GNN extracting node representations via learning suitable weights to fuse the information from multispaces.

8) **Geom-GCN** [33] is a geometric GNN aggregating neighbor information of nodes from a geometric perspective.

9) **BiTe-GCN** [7] is a text-rich GNN that obtains node representations by integrating word sequence structure.

10) **AS-GCN** [8] is a text-rich GNN that effectively integrates local and global semantics.

---

1https://zhangle18f.myweb.cs.uwindsor.ca/datasets/
2https://www.cs.cornell.edu/projects/dlink/datasets.html
3) Implementation Details: For all baselines, we adopt the default parameter settings released by their papers. For BiTe-GCN and AS-GCN, we initialize entity representations using pretrained 300-D GloVe representations [34]. For our model, we use Wikipedia anchors to align mentions extracted from the textual description of each document node to Wikidata5M [35], which is a newly proposed large-scale knowledge graph containing 4M entities and 21M fact triplets. We set the heterogeneous graph attention layer as 2, the learning rate of the Adam optimizer as 0.005, and the weight decay as 5e-4. In addition, the threshold of TagMe and similarity score (cosine similarity) are searched in {0.1, 0.2, 0.3, 0.4} and {0.5, 0.6, 0.7, 0.8, 0.9}, respectively. We set the activation function as LeakyReLU with a slope 0.2, and employ a dropout ratio of 0.5 for preventing overfitting. For fairly comparing all methods, we generate ten random partitions for training, validation and test.

B. Node Classification

On the node classification task, the objective is to calculate the labels of the remaining nodes on the premise of giving a fraction of node labels. Considering that the variance of graph-structured data may be quite large, we report the mean Accuracy and Macro-F1 along with the standard deviation of ten independent trials with different random seeds.

As presented in Table III, TeKo performs consistently the best across all four datasets. Specifically, in terms of Accuracy, TeKo achieves up to 3.33%, 0.81%, 0.55%, and 0.71% better accurate than the best baseline method on Hep-Small, Cora-Enrich, DBLP-Five, and Hep-Large, respectively. In terms of Macro-F1, TeKo is 2.70%, 0.34%, 0.55%, and 1.15% more accurate than the best baseline method on these four datasets. These results not only demonstrate the superiority of comprehensively mining textual semantics underlying text-rich networks via external knowledge, but also prove the significance of our new propagation principle that facilitates the information interaction between network structure and textual information. In addition, TeKo is significantly superior to vanilla GCN (i.e., 10.00%, 3.40%, 1.60%, and 2.78% relative improvements in Accuracy, and 10.39%, 3.55%, 1.55%, and 2.78% relative improvements in Macro-F1), which implies that TeKo is capable of making a well-balanced combination of both network structure and textual semantics within a text-rich network. Also of note, comparing with BiTe-GCN and AS-GCN which are also designed for text-rich networks, TeKo also has an obvious improvement, which further verifies the robustness of fully comprehending textual semantics for text-rich network representations with external knowledge.

C. Node Clustering

We also compare these methods on the task of node clustering. For each method, the learned document embeddings are used as the input of the $K$-Means algorithm, in which $K$ is equivalent to the number of clusters. Considering that the clustering performance is vulnerable to the initial center, we adopt the mean normalized mutual information (NMI) and adjusted rand index (ARI) in the range of $[-1, 1]$ as the evaluation metrics along with the standard deviation of ten splits.

As presented in Tables IV and V, TeKo performs the best across all four datasets. Specifically, TeKo outperforms BiTe-GCN and AS-GCN, which are also designed for text-rich networks, TeKo also has an obvious improvement, which further verifies the robustness of fully comprehending textual semantics for text-rich network representations with external knowledge.
Fig. 4. Visualization results of the learnt embeddings of (a) GCN, (b) GAT, (c) Geom-GCN, (d) AS-GCN, and (e) TeKo on the DBLP dataset. Different colors correspond to different ground truth categorical labels.

| Methods            | Hep-Small Accuracy | Hep-Small Macro-F1 | Cora-Enrich Accuracy | Cora-Enrich Macro-F1 | DBLP-Five Accuracy | DBLP-Five Macro-F1 | Hep-Large Accuracy | Hep-Large Macro-F1 |
|--------------------|-------------------|--------------------|----------------------|----------------------|---------------------|---------------------|---------------------|---------------------|
| TeKo               | 71.54 ± 3.88      | 70.76 ± 4.37       | 92.11 ± 0.86         | 91.38 ± 0.95         | 95.12 ± 0.52        | 94.74 ± 0.68        | 53.02 ± 1.03        | 53.01 ± 1.01        |
| - w/o Triplet      | 69.49 ± 3.33      | 69.06 ± 3.30       | 91.18 ± 1.89         | 90.86 ± 1.97         | 94.90 ± 0.64        | 94.54 ± 0.71        | 50.66 ± 1.19        | 49.83 ± 1.42        |
| - w/o Textual      | 73.33 ± 3.48      | 72.84 ± 4.03       | 90.30 ± 1.58         | 89.48 ± 1.60         | 94.99 ± 0.46        | 94.66 ± 0.54        | 50.94 ± 0.95        | 50.07 ± 0.93        |
| TeKo (CNet)        | 69.84 ± 3.27      | 69.52 ± 3.51       | 91.52 ± 1.48         | 91.16 ± 1.54         | 94.93 ± 0.53        | 94.59 ± 0.63        | 52.34 ± 1.12        | 52.26 ± 1.35        |
| TeKo (Sum)         | 71.36 ± 3.59      | 70.28 ± 3.75       | 91.85 ± 1.67         | 91.27 ± 1.84         | 95.03 ± 0.49        | 94.70 ± 0.58        | 52.17 ± 1.07        | 51.40 ± 1.23        |
| TeKo (Concat)      | 66.92 ± 2.68      | 66.63 ± 2.90       | 90.89 ± 1.36         | 90.18 ± 1.82         | 94.24 ± 0.47        | 93.78 ± 0.42        | 50.98 ± 1.01        | 50.37 ± 1.40        |

TABLE VI

COMPARISONS OF OUR TEKo WITH ITS FIVE VARIANTS ON NODE CLASSIFICATION IN TERMS OF ACCURACY (%) AND MACRO-F1 (%)

D. Visualization

To provide a more intuitive comparison, we take the DBLP dataset as an illustrative example to embed and visualize our proposed TeKo and four SOTA baselines (GCN, GAT, Geom-GCN as well as AS-GCN). We use t-SNE [36] to down-scale the learned node representations to a 2-D space, where different colors mean different labels. Therefore, an ideal visualization result is that nodes of the same classes (in the same color) should be close to each other.

From Fig. 4, we can find that neither the visualization results of GCN or GAT are satisfactory, since nodes with the same class are dispersed and nodes with different classes are mixed together. The results of Geom-GCN and AS-GCN are relatively better but the borders among different classes are still not so clear. Comparatively, the visualization of TeKo performs better, in which the learned representations have a higher intraclass similarity and form more discernible clusters.

E. Ablation Study

For further investigating the validity of each component in TeKo, we conduct ablation studies by removing one component at a time, that is, 1) TeKo of removing triplet representation of entities, named as w/o Triplet; 2) TeKo of removing the textual representation of entities, named as w/o Textual; 3) TeKo of employing ConceptNet instead of Wikidata5M to generate triplet representation of entities, named as TeKo (CNet); 4) TeKo of employing summation operator instead of gating mechanism to generate the knowledge-based entity representation, named as TeKo (Sum); as well as 5) TeKo of employing concatenation operator instead of gating mechanism to generate the knowledge-based entity representation, named as TeKo (Concat).

We take their comparison on node classification in terms of Accuracy and Macro-F1 as an example. As presented in Table VI, we have the following observation: 1) TeKo outperforms its five variants in most cases (except on Hep-Small), indicating the effectiveness of using both structured triplets and unstructured entity descriptions for entity representation together. 2) TeKo w/o Textual is in general better than TeKo w/o Triplet on three out of the four datasets, which implies the textual representation of entity plays a more vital role for fully comprehend textual semantics and promote its interaction with network structure. 3) Compared to TeKo (CNet), TeKo improves by an average of 0.79% (and 0.59%) on four datasets in Accuracy (Macro-F1), which illustrates the effectiveness of introducing Wikidata5M to enrich entity triplet representation. 4) Compared to TeKo w/o Triplet and TeKo w/o Textual, TeKo (Sum) performs the best on three out of the four datasets. Specifically, TeKo (Sum) is on average 1.04% (and 0.84%) and 0.21% (and 0.12%) more accurate than TeKo w/o Triplet and TeKo w/o Textual in Accuracy (and Macro-F1), which demonstrates the necessity of comprehensively considering and fusing two types of external knowledge in learning entity representation. 5) Compared to TeKo (Sum) or TeKo (Concat) that use summation or concatenation operators to fuse the triple representation and textual representation of an
entity, the improvement brought by TeKo that uses the gating mechanism is more significant, which illustrates the rationality and soundness of adaptively fusing these two representations for the given learning objectives. In addition, a number of analytical models [37], [38] related to the benefits of the gating mechanism (such as improving the trainability of nonconvex deep neural networks) further ensure the effectiveness of representation fusion.

F. Analysis of Embedding Initialization Methods

To further validate the effectiveness of introducing external knowledge to learn entity representation, we take BiTe-GCN and AS-GCN as an example, and initialize their word embedding using Word2Vec [39] and Glove, respectively, while keeping ours unchanged. As presented in Table VII, we can find that compare to BiTe-GCN and AS-GCN using Word2Vec (or Glove), TeKo improves by an average of 2.17% (or 1.69%) and 2.02% (or 1.43%) on four datasets in terms of Accuracy. This illustrates the effectiveness and importance of introducing the knowledge-based entity representation to comprehensively mine semantics underlying text-rich networks.

G. Parameter Analysis

We estimate the sensitivity of two important hyperparameters, that is, the threshold $\delta_{\text{tag}}$ of TagMe and the threshold $\delta_{\text{sim}}$ of entity edge similarity. We take the node classification on all the four text-rich datasets as an example and report the Accuracy and Macro-F1 values.

1) Analysis of $\delta_{\text{tag}}$: The threshold $\delta_{\text{tag}}$ determines the number of entities within the generated semantic network. We vary its value from 0.1 to 0.4 and the corresponding results are shown in Fig. 5. As the increase of this threshold $\delta_{\text{tag}}$, the performance presents the tendency of first rising and then declining. This makes sense, as a too small threshold of TagMe would introduce some noise entities, whereas a too large threshold would filter some informative entities and thus weaken the reciprocal guidance between textual semantics and network structure.

2) Analysis of $\delta_{\text{sim}}$: The threshold $\delta_{\text{sim}}$ determines the number of edges between entities, which represents the local word-sequence semantic structure underlying the corpus. We vary its value from 0.5 to 0.9 and the corresponding results are displayed in Fig. 6. With the increase of this threshold of entity edge similarity, the values of metrics, including Accuracy and Macro-F1, also increase first and then start to descend. It is probably because a few entity edges may lead to information loss and insufficient information propagation, while a large number of entity edges may propagate several useless information.

H. E-Commerce Searching Application

For further verifying the scalability of our new TeKo, we collect an e-commerce searching dataset and apply it for solving the problem of relevance matching, that is, predicting whether the current example pair (query and item) is relevant or not. The collected dataset contains million-scale queries or items, where the detailed statistics are illustrated in Table VIII.

1) External Knowledge in E-Commerce Searching: We use the category information to serve as the external knowledge in e-commerce search. Specifically, the category information includes four different levels of categories, i.e., $Cid_1$, $Cid_2$, and $Cid_3$. They present as a tree-shape structure as shown in Fig. 7. Adding this information is helpful to analyze the
semantics of the current query or items. For example, for an item whose title is “red mac 2020 (made in U.S.),” we cannot ensure whether it is an electronic product or a cosmetic product. But if we further know that its Cid1 is “Clothes & Cosmetics,” then we can easily infer that this is a lipstick.

2) Baselines: Since this relevance matching problem lies in the scope of natural language matching, we compare our new approach TeKo with seven state-of-the-arts in this topic and one variant of TeKo.

1) MV-LSTM [40] is a deep model which assigns the importance score of each local keyword using rich context information to match two sentences.

2) K-NRM [41] is a kernel-based semantic matching model that uses embedding layer, translation, and kernel pooling to capture the word-level interaction relationship between both sentences.

3) ARC-I [42] is a Siamese architecture that matches two sentences with their representations by using a multi-layer perceptron.

4) ARC-II [42] is an advanced version of ARC-I. Compared to ARC-I, ARC-II more focuses on the interaction relationship between two sentences.

5) MatchPyramid [43] employs hierarchical convolution to capture different-level matching patterns such as unigram contained in both sentences.

6) DUET [44] calculates the final relevance score by summing the scores of both representation and interaction-based embeddings.

7) BERT2DNN [45] is a data-driven model which adopts the techniques of transfer learning and knowledge distillation for search relevance.

8) TeKo w/o Knowledge is a variant of our proposed TeKo. It adopts the pretrained embeddings on 2 billion e-commerce corpus to replace external knowledge to learn entity representation.

3) Metrics: We employ six commonly used metrics for measuring the effectiveness of our proposed TeKo and baselines, including Area Under receiver operator characteristic Curve (AUC), Accuracy, Precision, F1-score, Recall, as well as False Negative Rate (FNR). The lower FNR means the better method, whereas other metrics are the opposite. In particular, AUC is the most important of these metrics.

4) Experimental Results and Analysis: As illustrated in Table IX, our TeKo is consistently better than all the baseline methods. Particularly, compared to the popular e-commerce searching algorithm BERT2DNN, which promotes result relevance using transfer learning and knowledge distillation, the superiority of TeKo is even up to 4.23% and 3.73% improvements in Accuracy and F1-score, respectively. These results not only demonstrate the superiority of our TeKo in capturing query-item pair correlation, but also shows the rationality of our adopting a more advanced way (i.e., utilize external knowledge to comprehend textual semantics underlying text-rich situations for high-level relevance matching). In addition, compared to TeKo w/o Knowledge that adopts the pretrained embeddings to learn entity representation, the improvement brought by TeKo that uses external knowledge is more significant, namely 1.59%, 2.02%, 2.43%, 2.32%, 2.67%, and 2.67% in AUC, Accuracy, Precision, F1-score, Recall, and FNR, respectively. This further verifies the effectiveness of introducing external knowledge to improve the performance of relevance matching in e-commerce searching scenarios.

V. RELATED WORK

We concisely discuss some closely relevant research works, including classical GNNs, text-rich GNNs, knowledge-enhanced GNNs, as well as GNNs for text analysis.

1) Classical GNNs: The GNNs have drawn considerable research interests owing to the powerful modeling ability of graph-structured data. For instance, Bruna et al. [46] designs the graph convolutional operation via introducing graph Laplacian into the Fourier domain. Defferrard et al. [47] further promote the efficiency using Chebyshev polynomial expansion. After that comes GCN [16], a simplified graph convolutional operation via introducing graph Laplacian. SPC-GNN [48] designs a self-paced co-training strategy that trains multiple GNNs using different representations of the same training data for semi-supervised node classification. Though these methods can be used for text-rich network

| Methods         | AUC(*) | Accuracy | Precision | F1-score | Recall | FNR   |
|-----------------|--------|----------|-----------|----------|--------|-------|
| MV-LSTM         | 0.8760 | 0.8069   | 0.8392    | 0.7256   | 0.7541 | 0.2459|
| ARC-I           | 0.7945 | 0.7392   | 0.8005    | 0.6329   | 0.6830 | 0.3170|
| K-NRM           | 0.8424 | 0.7899   | 0.7341    | 0.6019   | 0.7333 | 0.2667|
| ARC-II          | 0.8466 | 0.7913   | 0.6931    | 0.7002   | 0.7439 | 0.2561|
| MatchPyramid    | 0.8758 | 0.8278   | 0.7741    | 0.7451   | 0.7683 | 0.2317|
| DUET            | 0.8682 | 0.8156   | 0.8159    | 0.7253   | 0.7337 | 0.2663|
| BERT2DNN        | 0.8906 | 0.7896   | 0.8565    | 0.8362   | 0.8169 | 0.1831|
| TeKo w/o Knowledge | 0.9092 | 0.8319   | 0.8648    | 0.8735   | 0.8824 | 0.1176|
| w/o Knowledge   | 0.8933 | 0.8117   | 0.8405    | 0.8503   | 0.8557 | 0.1443|

Table IX: Comparisons on an e-commerce searching dataset, where (*) denotes the dominant metric. The lower FNR means the better method, whereas other metrics are opposite.
representations, they fail to put enough thought into the textual semantics (e.g., local word-sequence or global topic) underlying text-rich networks, which is particularly important for information propagation along network topology.

2) Text-Rich GNNs: Recently, much attention has been paid to text-rich network representations. HyperMine [49] is designed for discovering hypernymy from text-rich networks. NetTaxo [50] focuses on topic taxonomy construction, and integrates text data and network structures simultaneously. LTRN [51] designs a minimally-supervised categorization framework using personalized PageRank sampling from a text-rich network perspective. BiTe-GCN [7] uses bidirectional convolution of topology and features to depict original network structure and local word-sequence structure extracted from text, and on this basis, AS-GCN [8] takes the global topic semantic structure into account. However, these methods mainly focus on modeling semantic structures underlying network data space (including local word-sequence and/or global topics) to learn text-rich network representation, which fails to well comprehend the semantic content contained in text-rich network, and reason over complex concepts and relational paths, inevitably influencing the reciprocity between network topology and textual semantics.

3) Knowledge-Enhanced GNNs: As GNNs become the most eye-catching tools for node representations, several efforts have been made to combine external knowledge and GNNs to boost the performance of downstream tasks. KGCN [52] designs a knowledge graph recommendation method to capture users’ preferences. KGAT [53] presents a collaborative knowledge graph recommendation by explicitly modeling high-order semantic relations. More recently, RGPHAT [54] designs a two-level knowledge graph completion mechanism that effectively uses the inherent and valuable neighborhood information surrounding an entity. Caps-GNN [55] proposes a novel knowledge-enhanced model which generates personalized review by considering multilevel user preferences. CompareNet [56] utilizes topics and entities extracted from the text to enrich news representation, and then detect fake news by comparing with external knowledge entities. CAGE [57] designs a context-aware graph embedding method that introduces external knowledge graphs to capture semantic-level structural information among news articles, so as to improve article embeddings for news recommendations. However, how to utilize the guidance of external knowledge to facilitate text-rich network representations is still an area that needs to be explored urgently.

4) GNNs for Text Analysis: Another research line relevant to our work is to apply GNNs for analyzing text. Some latest studies also consider modeling short texts by incorporating additional information such as words or topics. For instance, Text GCN [58] improves the performance of text classification by generating a heterogeneous word document graph from the corpus. HGAT [25] further introduces topics and entities to enrich the short textual semantics. TensorGCN [59] considers the text contextual information by generalizing the GNN into a text graph tensor. TextING [60] treats each text as an individual graph and learns word interactions at the text level for inductive text classification. TextGTL [61] proposes a nonheterogeneous graph construction method that jointly considers different linguistic information, including semantics, syntax, and sequence context for text classification.

HGCN [62] proposes a novel hierarchical graph convolutional network, which uses a section GCN to model the macrostructure of the document and a word GCN to extract the fine-grained features of the document, for structured long document classification. In summary, the above GNNs mainly focus on establishing associations for independent texts via building a graph structure, which essentially lies in the scope of text analysis, rather than text-rich network representations.

VI. Conclusion

In this article, we present a new text-rich GNN, namely TeKo, which effectively integrates both network structure and textual semantics with guidance from external knowledge for text-rich network representations. In particular, we first integrate rich textual semantic structures into original text-rich networks by constructing a heterogeneous semantic network. We then gain a deeper insight into the textual semantics by introducing both structured knowledge triples and unstructured entity descriptions. Furthermore, we design a reciprocal principle for information propagation on the heterogeneous semantic network, which realizes a well-balanced combination of network structure and textual semantics, ultimately improving the quality of text-rich network representations. By incorporating the textual semantics, our model can also relieve the topological limitations of GNNs such as heterophily. Extensive experiments across various public networks as well as a large-scale JD e-commerce searching dataset demonstrate that TeKo outperforms state-of-the-arts in terms of text-rich network representation.

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 1, pp. 4–24, Jan. 2021.
[2] D. Jin, C. Huo, C. Liang, and L. Yang, “Heterogeneous graph neural network via attribute completion,” in Proc. Web Conf., Apr. 2021, pp. 391–400.
[3] D. Jin et al., “A survey of community detection approaches: From statistical modeling to deep learning,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 2, pp. 1149–1170, Feb. 2023.
[4] Q. Tan, N. Liu, X. Zhao, H. Yang, J. Zhou, and X. Hu, “Learning to hash with graph neural networks for recommender systems,” in Proc. Web Conf., Apr. 2020, pp. 1988–1998.
[5] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous graph neural network,” in Proc. SIGKDD, 2019, pp. 793–803.
[6] S. Abu-El-Haija et al., “MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing,” in Proc. ICML, vol. 97, 2019, pp. 21–29.
[7] D. Jin et al., “BiTe-GCN: A new GCN architecture via bidirectional convolution of topology and features on text-rich networks,” in Proc. 14th ACM Int. Conf. Web Search Data Mining, Mar. 2021, pp. 157–165.
[8] Z. Yu et al., “AS-GCN: Adaptive semantic architecture of graph convolutional networks for text-rich networks,” in Proc. IEEE Int. Conf. Data Mining (ICDM), Dec. 2021, pp. 837–846.
[9] L. Hu et al., “Text-graph enhanced knowledge graph representation learning,” Frontiers Artif. Intell., vol. 4, Aug. 2021, Art. no. 697856.
[10] R. Speer, J. Chin, and C. Havasi, “ConceptNet 5.5: An open multilingual graph of general knowledge,” in Proc. AAAI Conf. Artif. Intell., Feb. 2017, vol. 31, no. 1, pp. 4444–4451.
[11] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” in Proc. AAAI, 2015, pp. 2181–2187.
Zhizhi Yu received the B.S. and M.S. degrees from Jilin University, Changchun, China, in 2016 and 2019, respectively. She is currently pursuing the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China.

Her research interests are mainly related to graph machine learning and its application, including graph neural networks, heterogeneous information networks, text-rich networks, and e-commerce search.

Di Jin (Member, IEEE) received the Ph.D. degree in computer science from Jilin University, Changchun, China, in 2012.

He was a Research Scholar with DMG, UIUC, Champaign, IL, USA, during 2019–2020. He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. To date, he has published more than 100 research papers in top-tier journals and conferences, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON CYBERNETICS (TTCB), AAAI, IJCAI, NeurIPS, and WWW. His research interests include graph data mining and graph machine learning, especially on community detection, network embedding and GNNs.

Jianguo Wei received the M.S. degree from Tianjin University, Tianjin, China, in 2004, and the Ph.D. degree from the Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2007.

He is currently a Professor at the College of Intelligence and Computing, Tianjin University. He has devoted himself to the research of machine intelligence especially on speech signal processing and nature language processing.

Yunfei Wu (Member, IEEE) received the Ph.D. degree in computer science from the College of William and Mary, Williamsburg, VA, USA, in 2016.

He was a Research Staff Member at IBM Thomas J. Watson Research Center, Ossining, NY, USA, and led more than ten research scientist team for developing novel Graph Neural Networks methods and systems, which leads to the #1 AI Challenge Project in IBM Research. He was a Principal Scientist at JD.COM Silicon Valley Research Center, Mountain View, CA, USA, leading a team of more than 30 machine learning/natural language processing scientists and software engineers to build intelligent e-commerce personalization systems. He is an Engineering Manager with the Content and Knowledge Graph Group, Pinterest, New York, NY, where they are building the next generation Knowledge Graph to empower Pinterest recommendation/research systems across all major surfaces including Homefeed, Search, Ads, and so on. He has published one book (in GNNS) and more than 100 top-ranked conference and journal papers, and he is a co-inventor of more than 40 filed U.S. Patents.

Dr. Wu received multiple IBM Awards including three-time Outstanding Technical Achievement Award. Because of the high commercial value of his patents, he has received eight invention achievement awards and has been appointed as IBM Master Inventors, class of 2020. He was the recipients of the Best Paper Award and Best Student Paper Award of several conferences such as IEEE ICC ’19, AAAI workshop on DLGMA’20 and KDD workshop on DLG ’19. He has co-organized more than ten conferences (KDD, AAAI, IEEE BigData) and the Founding Co-Chair for Workshops of Deep Learning on Graphs (with AAAI’21, AAAI’20, KDD’21, KDD’20, KDD’19, and IEEE BigData’19) and Deep Learning on Graphs for Natural Language Processing (with ICLR’22 and NAACL’22). He has currently served as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and ACM TRANSACTIONS ON KNOWLEDGE DISCOVERY FROM DATA.

Jiawei Han is the Michael Aiken Chair Professor at the University of Illinois at Urbana-Champaign, Champaign, IL, USA. He co-authored book titled Data Mining: Concepts and Techniques (Morgan Kaufmann) has been adopted popularly as a textbook worldwide. His research interests include data mining, information network analysis, text mining, and database systems, and their applications.

Dr. Han is a fellow of ACM.