IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 22 July 2022, accepted 9 August 2022, date of publication 16 August 2022, date of current version 22 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3198941

RESEARCH ARTICLE

LEHAN: Link-Feature Enhanced Heterogeneous Graph Attention Network

JONGMIN PARK AND SUNGSU LIM, (Member, IEEE)
Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, South Korea

Corresponding author: Sungsu Lim (sungsu@cnu.ac.kr)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korean Government (MSIT) through the Artificial Intelligence Convergence Research Center, Chungnam National University, under Grant 2020-0-01441; in part of the Project titled ''Marine Digital AtoN Information Management and Service System Development (2/5)'' funded by the Ministry of Oceans and Fisheries, South Korea, under Grant 20210650; and in part by the BK21 FOUR Program by Chungnam National University Research Grant, in 2022.

ABSTRACT Graph Neural Networks (GNNs) have been studied extensively and have performed well in solving complex machine learning tasks in recent years. Many GNN-based approaches focused on representing homogeneous graphs with only a single type of nodes and links. However, many real-world networks are heterogeneous, involving various types of nodes and links. Existing GNN-based approaches for representing heterogeneous graphs only focused on node features and meta-paths, which often causes difficulties in reflecting link features to learn the graph representations. To overcome this limitation, we propose a Link-feature Enhanced Heterogeneous graph Attention Network (LEHAN) that focuses on the node and link features to represent heterogeneous graphs. LEHAN consists of the node attention block and the link attention block, where each block aggregates node features and link features by attention mechanism with meta-paths information. The extensive experimental evaluations show that LEHAN outperforms the state-of-the-art graph embedding algorithms in node classification and clustering on real-world heterogeneous graphs.

INDEX TERMS Graph neural networks, graph attention networks, heterogeneous graph embedding.

I. INTRODUCTION

In a heterogeneous graph, where various types of nodes are connected by links, how can we effectively represent the nodes? Which features can be considered to learn informative node representations?

Graphs are a ubiquitous data structure for describing structural and attribute information. One of the critical challenges of exploiting machine learning algorithms for graph data is to embed or encode graph elements (usually nodes) as low-dimensional vectors that summarize the graph structure. Machine learning problems can be solved efficiently by representing a graph in a low-dimensional space. In the past decade, various graph embedding models have been proposed using matrix factorization [1], [2], divergence minimization [3], autoencoders [4], random walks [5], [6], *etc*.

Advances in deep learning have led to the emergence of Graph Neural Networks (GNNs), which have become dominant and fast-growing techniques for learning with graph data [7]–[9]. Although existing GNN-based models lead to high performance in solving various downstream machine learning problems, most focus on homogeneous graphs.

Heterogeneous graphs are helpful for modeling complex systems in which different types of nodes interact with each other. A common framework for learning heterogeneous graphs is defining and using meta-paths that are composite relations between different types of nodes. Each meta-path captures the higher-order proximity among nodes, and the meta-path-based models are widely adopted in heterogeneous graph embedding models [10]–[12]. Recently, GNN-based heterogeneous graph embedding models have been

The associate editor coordinating the review of this manuscript and approving it for publication was Yin Zhang.

proposed by adopting a message-passing approach to model higher-order proximities using meta-paths [13]–[15].

However, GNN-based heterogeneous graph embedding models have a limitation in that they only focus on meta-path information indicated as the node features and node-type sequences. For a given graph, we can take or compute a link feature defined for each pair of nodes. The link features have been proven to be useful in some graph machine learning tasks, including link prediction [16]. However, most link features tend to be ignored in representation learning, while the link features indicate important information about the relationships between various types of nodes.

In this paper, we present a new model named Link-feature Enhanced Heterogeneous graph Attention Network (LEHAN), whose architecture contains the node and link attention blocks. These two attention blocks enable us to aggregate the node and link features to learn an effective representation of each node in a heterogeneous graph. Experiments on real-world heterogeneous graphs show that our proposed model achieves more accurate performance than existing heterogeneous graph embedding algorithms.

The contributions of this paper are summarized as follows:

- We propose LEHAN that learns the node representations using both the node and link features.
- We design the link attention block of LEHAN that effectively aggregate the information of links.
- We empirically show that LEHAN achieves higher performance than the state-of-the-art models.

II. RELATED WORKS

A. GRAPH NEURAL NETWORKS

Recent research to adopt convolution operations to graphs is drawing much attention and achieved outstanding performance in graph representation learning. Convolutional Graph Neural Network (ConvGNN) is divided into two parts: spectral-based GNNs and spatial-based GNNs [7].

Spectral-based GNNs, including ChebNet [17] and GCN [18], perform convolution operations in the Fourier domain of a graph. Spatial-based GNNs, including GraphSAGE [19] and Graph Attention Network (GAT) [20], perform convolution operations in the graph domain directly. All of the GNN-based models mentioned above are achieved high performance in various tasks (*e.g.*, node classification, node clustering, and link prediction). However, because they are designed to handle homogeneous graphs, they cannot fully represent the particular structures and semantic information in heterogeneous graphs.

B. HETEROGENEOUS GRAPH EMBEDDING

Heterogeneous graph embedding [21] is performed to represent graph elements in heterogeneous graphs as vectors in a low-dimensional space. Recent studies have proposed various heterogeneous graph embedding models. These models are divided into two parts: traditional graph embedding models and GNN-based models.

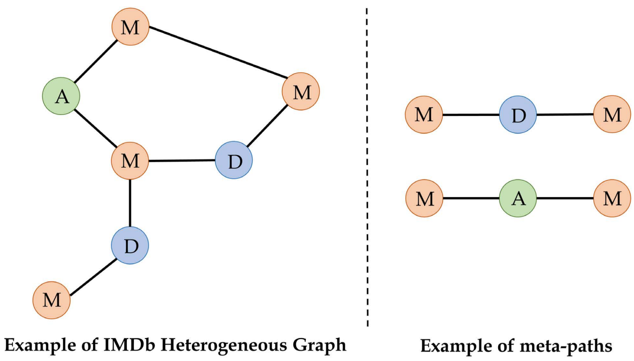

FIGURE 1. An example of a heterogeneous graph.

One of the most popular heterogeneous graph embedding models is Metapath2vec [12]. It analyzes random walks derived by meta-paths and adopts paths of nodes that passed random walks as input data of the skip-gram model for modeling different semantic data of relevant nodes. GNN-based models include HAN [14] and MAGNN [15]. HAN learns the representation of the target node by counting the meta-path-based neighbor nodes provided, and to train the weights of meta-path-based neighbor nodes, HAN adopts GAT. In this process, the intermediate nodes on the meta-path are ignored. MAGNN extends HAN, by considering both meta-path-based neighbor nodes and intermediate nodes included in the meta-path. MAGNN also adopts GAT.

All of the models mentioned above are achieved high performance in various tasks. However, they have difficulties reflecting the link features to a vector representation since they only focus on the node features and meta-paths.

Capturing the semantic relations between different types of nodes is effective in the representation learning process. Several studies have utilized heterogeneous graphs in recommendation systems to capture different semantic information. KCGN [22] proposes a relation-aware graph neural network to capture the multi-typed collaborative relations. ACKRec [23] constructs a heterogeneous graph to capture the different semantic information among different types of nodes. They achieved high performance by conducting heterogeneous graphs in recommendation systems. However, several real-world heterogeneous graphs are very sparse (e.g., tree-like networks [24] or long-circle-like networks [25]), so many existing recommendation models for heterogeneous graphs suffer from sparsity. To address this issue, we can adopt graph convolutional networks to leverage not only content information but also context information.

In this paper, considering these issues, we propose a graph attention network model that contains two attention blocks (*i.e.*, node attention block and link attention block) to capture both content information (*i.e.*, node features) and context information (*i.e.*, link features) for effective representation learning in heterogeneous graphs.

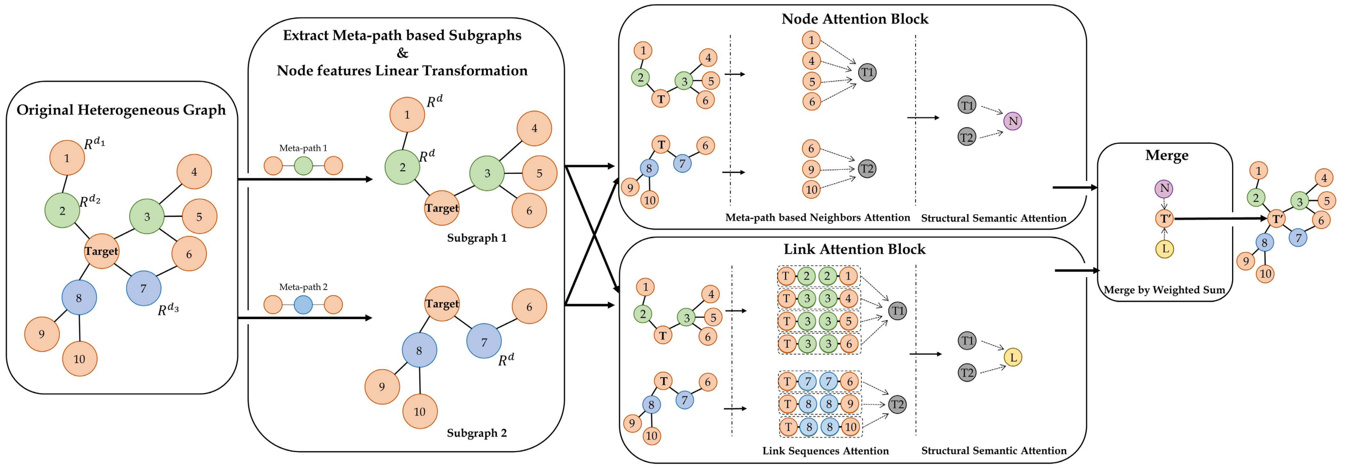

FIGURE 2. **An overview of the proposed LEHAN model.**

III. PRELIMINARIES

Our goal is to learn the effective representations of a given heterogeneous graph. In this section, we review basic concepts on heterogeneous graphs which are adopted in our proposed algorithm.

Definition 1 (Heterogeneous Graph): A heterogeneous graph is defined as a graph $G = (V, E)$, where V is a set of nodes and E is a set of links. A heterogeneous graph is associated with a node type mapping function $\Phi : V \rightarrow A$ and a link type mapping function $\Psi : E \rightarrow R$. A and R denote predefined sets of node types and link types, with $|A| + |R| > 2$.

Definition 2 (Meta-Path): A meta-path P is defined as a path $A_1 \xrightarrow{l_1} A_2 \xrightarrow{l_2} \cdots \xrightarrow{l_i} A_{i+1}$ (abbreviated as $A_1, A_2, \cdots, A_{i+1}$) which describes a composite relation $l = l_1 \circ l_2 \circ \cdots \circ l_i$ between nodes A_1 and A_{i+1}, where $\circ$ denotes the composition operator on relations.

Definition 3 (Meta-Path-Based Neighbors): Given a node i and a meta-path Φ in a heterogeneous graph, the meta-path-based neighbors $\mathcal{N}_i^{\Phi}$ of node i is defined as the set of nodes which connected by meta-path Φ. If meta-path Φ is symmetric, $\mathcal{N}_i^{\Phi}$ includes node i itself.

Figure 1 explains an example of a heterogeneous graph. The IMDb heterogeneous graph represents an online database data related to movies and television programs. There are three node types, a movie, an actor, and a director. In this example, a meta-path can be defined in various ways depending on the relationships between various types of nodes. For instance, an "movie-director-movie" meta-path indicates different movies directed by a specific director, and an "movie-actor-movie" meta-path indicates different movies acted by a specific actor.

IV. MODEL DESCRIPTION

In this section, we explain our new GNN-based model for heterogeneous graph embedding, which we call LEHAN. Our proposed model LEHAN consists of node attention block (Sect. IV-B), link attention block (Sect. IV-C), and merge phase (Sect. IV-D). The overall procedure is summarized in Figure 2.

First, we extract meta-path-based subgraphs from an original heterogeneous graph. Each subgraph is used as the input of two blocks that consist of our proposed model LEHAN. The node attention block aggregates features of meta-path-based neighbors, and the link attention block aggregates the features of links in a meta-path sequence. Then, both blocks aggregate the representations derived from each meta-path-based subgraph. Finally, LEHAN combines the representations from node and link attention blocks. This structure makes LEHAN reflects the representations of not only node features but also rich information from relations between nodes in heterogeneous graphs.

A. LINEAR TRANSFORMATION

For a heterogeneous graph, because of the heterogeneity of nodes, different types of nodes have different features, which are located in different feature spaces. Thus, it is important to project different types of node features into the same feature space before adopting a heterogeneous graph embedding model. Therefore, as shown in Equation 1, the linear transformation was conducted to project the feature vectors of different types of nodes to the same feature space. i is a node whose type is Φ. h_i is a node i's feature vector, and M_Φ is a projection matrix to project different types of node's feature vectors into the same feature space. h_i' is a projected feature vector of node i.

$$h_i' = M_\Phi \cdot h_i \tag{1}$$

By using the linear transformation, we can process different types of node features in our proposed model LEHAN without trouble.

Algorithm 1 Node Attention Block (Sect. IV-B)

Input: Heterogeneous graph $G = (V, E)$;
node features $\{h_i, \forall i \in V\}$;
meta-paths $\Phi = \{\Phi_1, \Phi_2, \cdots, \Phi_M\}$;
number of attention heads K;
Output: Node attention block embedding Z;

1: **for** $\Phi_m \in \Phi$ **do**
2: $h'_i \leftarrow M_{\Phi_m} \cdot h_i \; \{h_i, \forall i \in V\}$;
3: **for** $k = 1 \ldots K$ **do**
4: **for** $i \in V$ **do**
5: /* Node features attention (Sect. IV-B1) */
6: **for** $j \in \mathcal{N}_i^{\Phi_m}$ **do**
7: $e_{ij}^{\Phi_m} \leftarrow Att_{node}(h'_i, h'_j, \Phi_m)$;
8: $\alpha_{ij}^{\Phi} \leftarrow \textbf{softmax}(e_{ij}^{\Phi_m})$;
9: **end for**
10: $z_i^{\Phi_m} \leftarrow \sigma(\sum_{j \in N_i^{\Phi_m}} \alpha_{ij}^{\Phi_m} \cdot h'_j)$;
11: **end for**
12: $z_i^{\Phi_m} \leftarrow \|_{k=1}^{K} \sigma(\sum_{j \in N_i^{\Phi_m}} \alpha_{ij}^{\Phi_m} \cdot h'_j)$;
13: **end for**
14: /* Structural semantic attention (Sec. IV-B2) */
15: $d_{\Phi_m} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} p^T \cdot \tanh(A \cdot z_{\Phi_m} + bias)$;
16: $g_{\Phi_m} = \textbf{softmax}(d_{\Phi_m}) = \frac{exp(d_{\Phi_m})}{\sum_{\Phi_k \in \Phi} exp(d_{\Phi_k})}$;
17: $Z \leftarrow \sum g_{\Phi_m} \cdot z^{\Phi_m}$;
18: **end for**
19: **return** Z;

Algorithm 2 Link Attention Block (Sect. IV-C)

Input: Heterogeneous graph $G = (V, E)$;
node features $\{h_i, \forall i \in V\}$;
meta-paths $\Phi = \{\Phi_1, \Phi_2, \cdots, \Phi_M\}$;
number of attention heads K;
meta-path-based link sequences $\phi = \{\phi_1, \phi_2, \cdots, \phi_L\}$;
Output: Link attention block embedding R;

for $\Phi_m \in \Phi$ **do**
$h'_i \leftarrow M_{\Phi_m} \cdot h_i \; \{h_i, \forall i \in V\}$;
for $k = 1 \ldots K$ **do**
for $i \in V$ **do**
/* Meta-path Encoder (Sect. IV-C1) */
$h_i^{\phi} \leftarrow Encoder(\phi)$;
for $\phi_l \in \phi \; (\forall \phi_l \in \Phi_m)$ **do**
/* Link features attention (Sect. IV-C2) */
$w_i^{\phi_l} \leftarrow Att_{link}(h_i^{\phi_l}, \phi_l)$;
$\beta_i^{\phi_l} \leftarrow \textbf{softmax}(w_i^{\phi_l})$;
end for
$x_i^{\Phi_m} \leftarrow \sigma(\sum \beta_i^{\phi_l} \cdot h_i^{\phi_l})$;
end for
$x_i^{\Phi_m} \leftarrow \|_{k=1}^{K} \sigma(\sum \beta_i^{\phi_l} \cdot h_i^{\phi_l})$;
end for
/* Structural semantic attention (Sect. IV-C3) */
$D_{\Phi_m} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} q^T \cdot \tanh(B \cdot x_{\Phi_m} + bias)$;
$G_{\Phi_m} = \textbf{softmax}(D_{\Phi_m}) = \frac{exp(D_{\Phi_m})}{\sum_{\Phi_k \in \Phi} exp(D_{\Phi_k})}$;
$R \leftarrow \sum G_{\Phi_m} \cdot x^{\Phi_m}$;
end for
return R;

B. NODE ATTENTION BLOCK

Algorithm 1 shows the procedure of the node attention block. The node attention block is exploited to capture the node information from heterogeneous graphs [14]. The node attention block aggregates information from meta-paths. In this subsection, we explain how to aggregate information from each meta-paths.

1) NODE FEATURES ATTENTION

When $\mathcal{N}_i^{\Phi}$ is a set which contains the meta-path-based neighbors of node i for a given meta-path Φ, we adopt Graph Attention Network (GAT) to aggregate information from $\mathcal{N}_i^{\Phi}$ to node i. In the node attention block, the importance of each meta-path-based neighbor e_{ij}^{Φ} ($j \in \mathcal{N}_i^{\Phi}$) can be calculated by a standard attention mechanism, as shown in Equation 2.

$$e_{ij}^{\Phi} = Att_{node}(h'_i, h'_j, \Phi) = \sigma(a_{\Phi}^T \cdot [h'_i || h'_j]) \tag{2}$$

After calculating the importance of meta-path-based neighbors, we normalize e_{ij}^{Φ} to get the weight of each meta-path-based neighbors α_{ij}^{Φ}. In Equation 3, σ is activation function, $||$ is concatenation operator, and a_{Φ}^T is a node attention vector.

$$\alpha_{ij}^{\Phi} = \mathsf{softmax}(e_{ij}^{\Phi}) = \frac{exp(\sigma(a_{\Phi}^T \cdot [h'_i || h'_j]))}{\sum_{k \in \mathcal{N}_i^{\Phi}} exp(\sigma(a_{\Phi}^T \cdot [h'_i || h'_k]))} \tag{3}$$

After that, when the embedding vector of node i for the given meta-path Φ is z_i^{Φ}, it can be represented with the weight of $\mathcal{N}_i^{\Phi}$ and projected features of $\mathcal{N}_i^{\Phi}$.

$$z_i^{\Phi} = \sigma(\sum_{j \in \mathcal{N}_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot h'_j) \tag{4}$$

The embedding vectors of node features can be grouped by each meta-path Φ_m ($\Phi_m \in \Phi$), and the group of embedding vectors is denoted as z_{Φ_m} (line 12 of Algorithm 1).

2) STRUCTURAL SEMANTIC ATTENTION

In heterogeneous graphs, different meta-paths represent different semantic relationships. Therefore, the importance of different meta-paths is different in heterogeneous graphs. In order to learn about the importance of different meta-paths, we average the non-linear transformations of embedding vectors from the node features attention phase. Then we calculate the importance of each meta-path by adopting a standard attention mechanism. This process can be formulated as shown in the below equations.

$$d_{\Phi_m} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} p^T \cdot \tanh(A \cdot z_{\Phi_m} + bias) \tag{5}$$

$$g_{\Phi_m} = \mathsf{softmax}(d_{\Phi_m}) = \frac{exp(d_{\Phi_m})}{\sum_{\Phi_k \in \Phi} exp(d_{\Phi_k})} \tag{6}$$

$$Z = \sum_{\Phi_m \in \Phi} g_{\Phi_m} \cdot z_{\Phi_m} \tag{7}$$

C. LINK ATTENTION BLOCK

Algorithm 2 shows the procedure of the link attention block. The link attention block aggregates information from links contained in a meta-path. In this subsection, we explain how to aggregate information of links of meta-paths.

1) META-PATH ENCODER

Meta-path encoder [15] is adopted to convert meta-path-based link sequences to a single vector named meta-path instance. In this paper, a meta-path instance is regarded as a single vector of link features that are included in the meta-path. As the meta-path encoder, we adopt a relational rotation encoder proposed by RotatE [26], which is proposed to process knowledge graphs. By adopting RotatE as a meta-path encoder, we can represent relations between different links included in the meta-path to vector. In this phase, we can also consider the sequential structure of the meta-path.

Let $\Phi = A_1 \xrightarrow{l_1} A_2 \xrightarrow{l_2} \cdots \xrightarrow{l_i} A_{i+1}$ be a meta-path. When $t_0 = l_1$ and $t_i = l_i$, the relation between two links l_{i-1} and l_i is $\mathcal{R}_i$ and r_i is a relation vector of $\mathcal{R}_i$. The relational rotation encoder adopted as a meta-path encoder is formulated as shown in the below equations.

$$o_0 = h'_{t_0} = h'_{l_1} \tag{8}$$

$$o_i = h'_{t_i} + o_{i-1} \odot r_i \tag{9}$$

$$h_\phi = \frac{o_n}{n+1} \tag{10}$$

In Equation 9, h'_{t_i} and r_i are both complex vectors, $\odot$ is the element-wise product of vectors. In addition, in Equation 10, h_ϕ is a single vector that is represented by the encoding of links features which are included in a given meta-path-based link-sequence ϕ. In the link attention block, h_ϕ is used as an input of GAT.

2) LINK FEATURES ATTENTION

When h_ϕ is a single vector represented by the encoding of links features that are included in meta-path-based link sequences ϕ, which include node i, we adopt GAT to aggregate information from h_ϕ. In the link attention block, the importance of every single vector of link features w_ϕ is calculated by a standard attention mechanism, as shown in Equation 11.

$$w_\phi = Att_{link}(h'_i, h_\phi) = \sigma(b_\phi^T \cdot [h'_i || h_\phi]) \tag{11}$$

In Equation 11, b_ϕ^T is a link attention vector and $\phi = \{\phi_1, \ \phi_1, \ \cdots \phi_M\} \subset \Phi$ is a set of link sequences which include node i and ϕ is a subset of given meta-path Φ. That is, ϕ is a set of link sequences which connected with node i. After calculating the importance of single vector of meta-path-based link sequence w_ϕ, we normalize them to get the weight of each single vector of meta-path-based link sequence β_ϕ.

$$\beta_{\phi_m} = \mathsf{softmax}(w_{\phi_m}) = \frac{exp(\sigma(b_{\phi_m}^T \cdot [h'_i || h_{\phi m}]))}{\sum_{\phi_k \in \phi} exp(\sigma(b_{\phi_k}^T \cdot [h'_i || h_{\phi k}]))} \tag{12}$$

After that, when the embedding vector for a single vector of link features for a given meta-path Φ that includes node i is x_i^Φ, it can be represented with β_ϕ and h_ϕ, as shown in Equation 13.

$$x_i^\Phi = \sigma(\sum_{\phi_k \in \phi} \beta_{\phi_k} \cdot h_{\phi k}) \tag{13}$$

3) STRUCTURAL SEMANTIC ATTENTION

Similarly to Sect. IV-B2, we average the non-linear transformations of embedding vectors from link features attention phase. Then we calculate the importance of each meta-path by adopting an attention mechanism. This process can be formulated as shown below equations. The embedding vector of the link feature can be grouped given meta-path Φ_m ($\Phi_m \in \Phi$), and the group of embedding vector denoted as x_{Φ_m}.

$$D_{\Phi_m} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} q^T \cdot \tanh(B \cdot x_{\Phi_m} + bias) \tag{14}$$

$$G_{\Phi_m} = \mathsf{softmax}(D_{\Phi_m}) = \frac{exp(D_{\Phi_m})}{\sum_{\Phi_k \in \Phi} exp(D_{\Phi_k})} \tag{15}$$

$$R = \sum_{\Phi_m \in \Phi} G_{\Phi_m} \cdot x_{\Phi_m} \tag{16}$$

D. MERGE PHASE

Embedding vectors from the node attention blocks and the link attention blocks are merged in this merge phase. Because the node features and the link features have different importance, we average vectors which are non-linear transformations of two embedding vectors. Then, we calculate the importance of each embedding vector by adopting an attention mechanism. This way, LEHAN can learn about the importance of each embedding from each attention block. In the below equations, B is a set of two attention blocks (*e.g.*, the node and link attention blocks), and E is a set of embeddings from the two attention blocks.

$$U_b = \frac{1}{2} \sum_{b \in B} o^T \cdot tanh(O \cdot E_b + bias) \tag{17}$$

$$I_b = \mathsf{softmax}(U_b) = \frac{exp(U_b))}{\sum_{k \in B} exp(U_b))} \tag{18}$$

$$P = \sum_{b \in B} I_b \cdot E_b \tag{19}$$

E. MODEL LEARNING

We apply semi-supervised learning to train the node representation based on heterogeneous graphs after deriving an embedding for a particular node using the process described above. For semi-supervised learning, we can adopt cross-entropy as a loss function. Cross-entropy can be formulated as shown in Equation 20.

$$\mathcal{L} = -\sum_{v \in V_L} \sum_{k=1}^{K} (y_v[k] \cdot \log P_v[k]) \tag{20}$$

Here, V_L is the set of labeled nodes, and K is the number of classes. y_v is the one-hot encoded label vector of node v, and P_v is a vector predicting the label probabilities of v.

V. EXPERIMENTAL RESULTS

In this section, we examine the performance of our proposed model LEHAN. We extensively tested LEHAN on two widely-used heterogeneous graphs from different domains. We compared the performance of eight algorithms including LEHAN. We show experimental results on the node clustering and classification tasks using the graph datasets.

A. DATASETS

Table 1 lists the real-world heterogeneous graphs used for our experiments. These datasets are from [15].

TABLE 1. Real-world heterogeneous graphs for experiments.

Name	# nodes	# links	Meta-paths
IMDb	# movie (M): 4,278 # director (D): 2,081 # actor (A): 5,257	# M-D: 4,278 # M-A: 12,828	MDM MAM DMD DMAMD AMA AMDMA
DBLP	# author (A): 4,057 # paper (P): 14,328 # term (T): 7,723 # venue (V): 20	# A-P: 19,645 # P-T: 85,810 # P-V: 14,328	APA APTPA APVPA

- **IMDb**: The IMDb dataset includes online database data related to movies and television programs. It consists of 4,278 movie type nodes (M), 2,081 director type nodes (D), and 5,257 actor type nodes (A) from original dataset. The movie type nodes were labeled as three classes (*i.e.*, action, comedy, and drama) according to movie genres. Moreover, the meta-path set consists of six meta-paths {MDM, MAM, DMD, DMAMD, AMA, AMDMA}.
- **DBLP**: The DBLP dataset includes data from a list of research papers on computer science. It consists of 4,057 author type nodes (A), 14,328 paper type nodes (P), 7,723 term type nodes (T), and 20 venue type nodes (V) from original dataset. The author type nodes were labeled as four classes (*i.e.*, database, data mining, artificial intelligence, and information retrieval) according to authors' research fields. Additionally, the meta-path set consists of three meta-paths {APA, APTPA, and APVPA}.

B. BASELINES

We compared our proposed model LEHAN with several state-of-the-art graph embedding models. These embedding models are divided into (i) unsupervised learning models including node2vec [6], metapath2vec [12], and HERec [27] and (ii) semi-supervised learning models including GCN [18], GAT [20], HAN [14], and MAGNN [15].

- **node2vec**: node2vec is a traditional embedding model for homogeneous graphs, which was developed based on the generalization of DeepWalk. To apply this model to heterogeneous graphs, various types of nodes in a dataset were united and used as a single type.
- **metapath2vec**: metapath2vec is a traditional embedding model for heterogeneous graphs, which generate node embeddings by using meta-path guided random walks as inputs of skip-gram models. metapath2vec only uses a single meta-path which is pre-defined by the user. So we report only the best results which experimented for all meta-paths.
- **HERec**: HERec is a traditional embedding model for heterogeneous graphs based on a random walk. HERec converts a heterogeneous graph to a homogeneous graph based on neighbor nodes on meta-paths. Then HERec applies the DeepWalk model to the meta-path-based homogeneous graph to train the embedding of the target-type nodes.
- **GCN**: GCN is a graph neural network model for homogeneous graphs. GCN performs convolution operations on graphs in the Fourier domain. We combined and used various types of nodes in meta-path-based subgraphs as single node types. We report the best results from all meta-paths.
- **GAT**: GAT is a graph neural network model for homogeneous graphs. GAT performs convolution operations on graphs in the Spatial domain by adopting an attention mechanism. Similar to GCN, We combined and used various types of nodes in meta-path-based subgraphs as single node types and report the best results from all meta-paths.
- **HAN**: HAN is a graph neural network model for heterogeneous graphs. HAN trains node embedding using meta-path-based neighbor nodes, except for intermediate nodes, which are included in meta-paths, and generates node embedding by adopting an attention mechanism.
- **MAGNN**: MAGNN is a graph neural network model for heterogeneous graphs. MAGNN trains node embeddings using meta-path neighbor nodes, including intermediate nodes, which are included in meta-paths, and generates node embedding by adopting an attention mechanism.

Remark that node2vec, GCN, and GAT are designed for homogeneous graphs, while the others are designed for heterogeneous graphs.

TABLE 2. Experiment results (%) on the IMDb and DBLP datasets for the node classification task.

Dataset	Metric	Train%	Unsupervised			Semi-supervised				
			node2vec	Metapath2vec	HERec	GCN	GAT	HAN	MAGNN	LEHAN
IMDb	Macro-F1	20%	48.87	45.98	45.87	52.81	53.65	56.17	59.32	**61.03**
		40%	50.12	47.35	46.72	53.97	55.37	56.21	60.12	**61.31**
		60%	51.87	47.89	46.96	54.11	56.56	57.13	60.72	**61.70**
		80%	51.94	49.77	47.68	54.74	57.24	58.44	61.14	**62.20**
	Micro-F1	20%	49.91	47.13	46.31	52.77	53.63	56.36	58.98	**61.05**
		40%	52.10	48.06	47.78	53.69	54.87	57.14	60.48	**61.40**
		60%	52.74	49.83	48.21	54.32	56.32	58.39	60.87	**61.72**
		80%	52.78	50.36	49.07	54.58	57.21	59.32	61.51	**62.23**
DBLP	Macro-F1	20%	86.84	88.49	90.83	87.97	91.10	91.58	93.14	**93.52**
		40%	57.97	89.31	91.41	88.75	91.23	92.08	93.31	**93.83**
		60%	88.65	89.97	92.07	89.42	91.74	92.27	93.49	**94.11**
		80%	88.83	90.70	92.33	89.87	91.92	92.48	94.12	**94.13**
	Micro-F1	20%	87.14	89.62	91.47	88.38	91.51	92.32	93.58	**93.96**
		40%	88.56	90.15	92.11	89.13	91.84	92.47	93.69	**94.26**
		60%	88.98	90.87	92.64	89.58	92.13	93.12	94.10	**94.37**
		80%	89.41	91.24	92.74	90.18	92.38	93.24	**94.45**	94.42

TABLE 3. Experiment results (%) on the IMDb and DBLP datasets for the node clustering task.

Dataset	Metric	Unsupervised			Semi-supervised				
		node2vec	Metapath2vec	HERec	GCN	GAT	HAN	MAGNN	LEHAN
IMDb	NMI	5.34	0.78	0.43	7.48	7.95	10.83	**15.67**	14.81
	ARI	6.12	0.31	0.14	7.81	8.76	11.24	**16.72**	14.12
DBLP	NMI	76.87	73.84	68.85	74.52	71.17	77.57	80.78	**81.42**
	ARI	80.92	78.29	72.36	77.63	76.89	83.06	85.87	**86.58**

C. EXPERIMENT SETUP

For unsupervised models, we set the window size to 5, walk length to 100, walks per node to 40, and the number of negative samples to 5 for unsupervised models. In our experiments, for a fair comparison, all unsupervised models use the same hyperparameter values. For semi-supervised (or GNN-based) models, including LEHAN, we set the dropout rate to 0.5, and we use the Adam optimizer with a learning rate of 0.005 and weight decay (L2 penalty) of 0.001. The performance of each model was measured ten times, and the highest performance was selected. As in the unsupervised models, all semi-supervised models used the same hyperparameter values.

D. EXPERIMENT RESULTS

We conducted a set of experiments to examine the superiority of the proposed model LEHAN. The aim of the first experiment is to evaluate the performance of node classification. The aim of the second one is to evaluate the accuracy of node clustering.

1) NODE CLASSIFICATION

We conducted the classification of labeled nodes among embedded nodes using Support Vector Machine (SVM) to experiment with the node classification based on IMDb and DBLP datasets. The performance of each model was measured ten times, and the highest performance was selected. Table 2 presents the Macro-F1 and Micro-F1 scores measured in the node classification tasks based on SVM. The ratio of the training data was adjusted in the range from 20% to 80%. Moreover, because the semi-supervised learning models already learned training and validation data, only the nodes of the test data were used as input data for SVM. In the IMDb dataset, LEHAN achieved higher performance by $1 \sim 2\%$ than MAGNN and other competitors. Moreover, in the DBLP dataset, LEHAN showed similar or slightly higher performance than MAGNN and other competitors.

2) NODE CLUSTERING

We conducted clustering of labeled nodes among embedded nodes using a k-means clustering algorithm to experiment on the node clustering based on IMDb and DBLP datasets. The performance of each model was measured ten times, and the highest performance was selected. Table 3 presents the Normalized Mutual Information (NMI) [28] and Adjusted Rand Index (ARI) [29] measured in the node clustering tasks. In IMDb dataset, LEHAN shows the best NMI and ARI values. In DBLP dataset, although LEHAN showed lower performance by approximately 1% than MAGNN but also achieved higher performance than the other competitors.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a Link-feature Enhanced Heterogeneous graph Attention Network (LEHAN) for heterogeneous graph embedding by using not only node features but also link features. LEHAN represents various meta-paths, structural properties, and semantic information of heterogeneous graphs rather than depending on a single pre-defined meta-path.

In order to represent information on nodes and links, LEHAN consists of two types of blocks. In the node attention block, we can reflect neighbors' features to the center node by depending on the weight of neighbors. In the link attention block, we can reflect link features included in meta-paths to the center node depending on the weight of the meta-paths connected to the center node. The resulting vectors from the two attention blocks are merged. This structure makes LEHAN can represent node to vector by the weights of nodes and links.

Experiments on heterogeneous graph datasets show that LEHAN outperforms various state-of-the-art models. From the results of empirical experiments, we realize that in the heterogeneous graph representation learning, the link features are as crucial as node features.

We plan to develop a theoretical analysis of the effect of link attention in the future. Also, our proposed model LEHAN is expected to be effectively applied to analyze smart home networks and E-commerce platforms represented as heterogeneous graphs and also to recommendation systems.

REFERENCES

[1] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 891–900.

[2] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1105–1114.

[3] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.

[4] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1225–1234.

[5] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.

[6] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[7] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2021.

[8] W. Zhang, Y. Zhang, L. Xu, J. Zhou, Y. Liu, M. Gu, X. Liu, and S. Yang, "Modeling IoT equipment with graph neural networks," *IEEE Access*, vol. 7, pp. 32754–32764, 2019.

[9] N. A. Asif, Y. Sarker, R. K. Chakrabortty, M. J. Ryan, M. H. Ahamed, D. K. Saha, F. R. Badal, S. K. Das, M. F. Ali, S. I. Moyeen, M. R. Islam, and Z. Tasneem, "Graph neural network: A comprehensive review on non-Euclidean space," *IEEE Access*, vol. 9, pp. 60588–60606, 2021.

[10] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-k similarity search in heterogeneous information networks," in *Proc. PVLDB*, 2011, pp. 992–1003.

[11] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1165–1174.

[12] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 135–144.

[13] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. ESWC*, 2018, pp. 593–607.

[14] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, May 2019, pp. 2022–2032.

[15] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. Web Conf.*, Apr. 2020, pp. 2331–2341.

[16] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 553, Sep. 2020, Art. no. 124289.

[17] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NIPS*, 2016, pp. 3844–3852.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, pp. 1–14.

[19] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*, 2017, pp. 1024–1034.

[20] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018, pp. 1–12.

[21] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, "Heterogeneous network representation learning: A unified framework with survey and benchmark," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 21, 2020, doi: 0.1109/TKDE.2020.3045924.

[22] C. Huang, H. Xu, Y. Xu, P. Dai, L. Xia, M. Lu, L. Bo, H. Xing, X. Lai, and Y. Ye, "Knowledge-aware coupled graph neural network for social recommendation," in *Proc. AAAI*, 2021, pp. 4115–4122.

[23] J. Gong, S. Wang, J. Wang, W. Feng, H. Peng, J. Tang, and P. S. Yu, "Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 79–88.

[24] K.-K. Shang, T.-C. Li, M. Small, D. Burton, and Y. Wang, "Link prediction for tree-like networks," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 29, no. 6, Jun. 2019, Art. no. 061103.

[25] K.-K. Shang and M. Small, "Link prediction for long-circle-like networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 105, no. 2, Feb. 2022, Art. no. 024311.

[26] X. Zhou, Y. Yi, and G. Jia, "Path-RotatE: Knowledge graph embedding by relational rotation of path in complex space," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Jul. 2021, pp. 905–910.

[27] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[28] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *JMLR*, vol. 3, pp. 583–617, Dec. 2003.

[29] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, 1985.

JONGMIN PARK received the B.S. degree in computer science and engineering from Chungnam National University, Daejeon, South Korea, in 2021, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. His research interests include data mining and machine learning with graphs.

SUNGSU LIM (Member, IEEE) received the B.S. and M.S. degrees in mathematical science and the Ph.D. degree in knowledge service engineering from KAIST, Daejeon, South Korea, in 2009, 2011, and 2016, respectively. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, Daejeon. His research interests include mining and modeling large-scale networks and their theoretical aspects.