IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 October 2023, accepted 9 October 2023, date of publication 12 October 2023, date of current version 19 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3324046

RESEARCH ARTICLE

Companion Animal Disease Diagnostics Based on Literal-Aware Medical Knowledge Graph Representation Learning

VAN THUY HOANG[1], THANH SANG NGUYEN[1], SANGMYEONG LEE[2], JOOHO LEE[3], LUONG VUONG NGUYEN[4], AND O-JOUN LEE[1], (Member, IEEE)

[1]Department of Artificial Intelligence, The Catholic University of Korea, Bucheon, Gyeonggi 14662, Republic of Korea
[2]Department of Physics, The Catholic University of Korea, Bucheon, Gyeonggi 14662, Republic of Korea
[3]School of Computer Science and Information Engineering, The Catholic University of Korea, Bucheon, Gyeonggi 14662, Republic of Korea
[4]Department of Artificial Intelligence, FPT University, Da Nang 550000, Vietnam

Corresponding author: O-Joun Lee (ojlee@catholic.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2022R1F1A1065516 and Grant 2022K1A3A1A79089461; in part by the Research Fund, 2023 of The Catholic University of Korea, under Grant M-2023-B0002-00088; and in part by the Research Fund of IntoCNS under Grant M-2022-D0827-00001.

ABSTRACT Knowledge graph (KG) embedding has been used to benefit the diagnosis of animal diseases by analyzing electronic medical records (EMRs), such as notes and veterinary records. However, learning representations to capture entities and relations with literal information in KGs is challenging as the KGs show heterogeneous properties and various types of literal information. Meanwhile, the existing methods mostly aim to preserve graph structures surrounding target nodes without considering different types of literals, which could also carry significant information. In this paper, we propose a knowledge graph embedding model for the efficient diagnosis of animal diseases, which could learn various types of literal information and graph structure and fuse them into unified representations, namely LiteralKG. Specifically, we construct a knowledge graph that is built from EMRs along with literal information collected from various animal hospitals. We then fuse different types of entities and node feature information into unified vector representations through gate networks. Finally, we propose a self-supervised learning task to learn graph structure in pretext tasks and then towards various downstream tasks. Experimental results on link prediction tasks demonstrate that our model outperforms the baselines that consist of state-of-the-art models.

INDEX TERMS Medical knowledge graph embedding, disease diagnosis, companion animal disease.

I. INTRODUCTION

Identifying animal diseases early is important to prevent and control further companion animal diseases and spread. For the diagnosis of animal diseases, pet owners mostly rely on professional veterinarians who possess general medical knowledge. However, the lack of high-level experts and timelines could not be guaranteed, resulting in a great financial loss. Therefore, there is a critical need to find efficient methods to assist experts in efficiently diagnosing animal diseases. Furthermore, sick animals provide historical diagnosis records from different sources, which could bring valuable information for expert systems to explore latent knowledge.

Recently, knowledge graphs (KGs) have shown the power to solve important tasks in the biomedical area, especially in animal disease diagnostics. KGs could represent natural relations of entities that are extracted from electronic medical records (EMRs), which then be used to explore useful knowledge [1], [2], [3], [4], [5], [6], [7], [8]. Several object information in EMRs, i.e., edge, gender, and treatments, are defined as entities, and edges denote the relations of entities. By using KGs to represent entities and their relations, learning representations of KGs could assist as an auxiliary role in providing expert decision support. Accordingly, various graph embedding methods have been proposed

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

to learn different types of relations among entities in an MKG [4], [9]. The learned representations are then utilized to solve downstream tasks, such as disease diagnosis and medicine recommendation [4], [10], [11].

Most of the graph-based methods have been proposed to mainly learn the graph structure in diagnosing animal diseases while ignoring literal information [12], [13]. Since knowledge graphs have heterophily properties and different types of relations, the learned representations thus could be limited to capturing the implicit and complex relations in KGs. For example, literal information in EMRs, i.e. symptoms, text description, and doctor's advice for the treatments, could also benefit the models to predict animal diseases [14], [15], [16]. Consequently, the performance of the models could be reduced as the models overlooked literal information.

Most Graph neural networks (GNNs) have shown great power in learning representations to address homophily graphs [17], [18]. However, a few studies apply GNN-based models to solve the knowledge graph problem [10], [11]. Applying GNNs to knowledge graphs is a challenging task because knowledge graphs have heterogeneous properties, complex relations, and different types of entities. Moreover, most GNN models aim to directly solve classification and link prediction tasks, which is not in an end-to-end manner. Considering the limitations of existing GNNs, we believe it is critical to develop a GNN-based model that could learn different types of relations and complex relations efficiently and in an end-to-end manner. There have been several studies [16], [19] that leverage literal information to capture the semantic relations and improve the learned representations, two limitations of the existing models restrict their ability to represent KG structure. First, some methods treat various entity attributes and their relations equally, which could not be sufficient for learning important features of different entities [20]. Second, these methods mainly learn textual information through several deep learning models, i.e., RNNs and LSTM, as auxiliary modules to capture the literal information [19]. The entity attributes and literal information are then learned independently, which could not bring the expressiveness of the graph-based model to handle heterogeneous properties.

To address this limitation, we represent LiteralKG, a KG representation learning model that could learn both different types of literal information and graph structure and then fuse them into unified representations. In particular, we first transform entities and their attributes into unified representations by using a gate function [21]. Since the different attributes are formed from different types of information, such as numerical and textual values, such fusion layers could benefit for representing unified vectors and then embedding models [22]. After composing literal enriched embedding vectors, the representations could be encoded and obtained from attentive embedding propagation layers [23], [24]. We propose an embedding propagation mechanism that could learn recursively neighbourhood information with an attention mechanism. The propagation mechanism could benefit the model by capturing high-order structural information with complex relations, which could be suitable for knowledge graphs. Note that our propagation mechanism differs from the original GNN aggregators and GAT model [23]. We first apply different GNN aggregators to capture the neighbourhood information surrounding target entities to learn the vector representations. Then, the embedding propagation mechanism captures the coefficients across various triplets in our knowledge graph. We believe that our proposed model could capture the relations between different entities with the learned attentional weights. It means that we only use GNN aggregators to aggregate the neighbourhood information of each entity. Then, the embedding propagation mechanism could capture the complex relations between head and tail entities attentively. Furthermore, we adopt a self-supervised learning task that could learn graph structures from pretext tasks without using any label information to generate learned representations and then use the representations for downstream tasks, such as link prediction [25].

In particular, we focus on several research questions:

- RQ1. Could fusing different types of literal information and entity relations benefit the representation learning to discover explicit and complex relations in KGs?
- RQ2. Which characteristics of encoders could benefit the model in learning various types of features and complex relations in KGs?
- RQ3. Could the pre-training model with pre-training tasks generate effective representations that could then benefit downstream tasks?

To answer RQ1, we aim to conduct experimental cases to explore which types of attributes could benefit the learned representations and contribute to the overall performance of LiteralKG. Since there are two types of literal features, i.e., numerical and textual information, it is critical to investigate combinations between different forms of features that can contribute to the effective performance of LiteralKG. For RQ2, we aim to investigate the performance of our model on different types of GNN aggregators. It is worth noting that most GNN aggregators are initially designed for homogeneous and simple graphs. However, applying GNN aggregators to KGs could be different as KGs show heterophily properties and complex relations. Furthermore, we aim to investigate which aggregators could be appropriate and show a powerful aggregation function to capture the complex relations in our data. For RQ3, we investigate the efficiency of the pre-trained model trained by optimizing a triplet loss function. Pre-training tasks could learn representations to capture underlying relations and structures without using labelled information. We aim to investigate whether learned representations could benefit downstream tasks [26], [27]. Therefore, the pre-training model is expected to bring more efficiency in extracting complex relations and benefit downstream tasks.

The contributions of this paper are summarized as follows:

(i) We construct a medical knowledge graph that comprises 595,172 entities and 16 relation types from various EMRs.

(ii) We propose LiteralKG, a knowledge graph embedding model that could learn different types of literal information and graph structure and then fuse them into unified representations with an attentive propagation mechanism.

(iii) We propose a self-supervised learning task that could learn the graph structure from pretext tasks to generate representations, and then the pre-trained model is used for downstream tasks to predict animal diseases.

(iv) The experimental results on the KG with different types of GNN aggregators and residual connection and identity mapping show the superiority of LiteralKG over baselines.

The rest of this paper is organized as follows. Section II presents literature reviews of existing methods to solve the above problems. Section III describes our proposed strategies to construct a medical KG from EMRs. The methodology of LiteralKG is presented in section IV. Section V shows the experimental results and analysis. Section VI is the conclusion and future work.

II. RELATED WORK

Over recent years, several graph-based methods have been proposed to handle medical records through learning graph structure [4], [9], [28], [29]. For example, several studies [14], [15] use translation-based methods, such as TransE [30] and TransR [31], to map entities and relations into latent space and then predict diseases. PrTransX [9] enhances translation-based methods, such as TransE [30], TransH [32], TransR [31], TransD [33], or TranSparse [34] by optimizing triplet probability into a scoring function and the margin-based loss function to enrich representations. Gong et al. [4] have proposed a model representing diseases, medicines, and patient data in EMRs by utilizing a KG triplet loss function. However, most existing models aim to learn entity representations without considering the literal information, i.e. symptoms and doctor's advice, which could carry significant information for learning representations [35]. In contrast, our model could learn different side information types, such as numeric and literal features.

Several studies have been proposed to capture entity features and literal information to enrich learned representations. For example, Tay et al. [36] introduce AttrNet model to learn the entities and their relations in triplets through the combination with attribute features. Wu et al. [37] have proposed TransEA, which can represent the numerical features of entities and learn the attribute triplet based on an attribute score. Kristiadi et al. [21] have proposed the LiteralE model to learn the numerical and textual features through linear transformation and optimize a triplet scoring function.

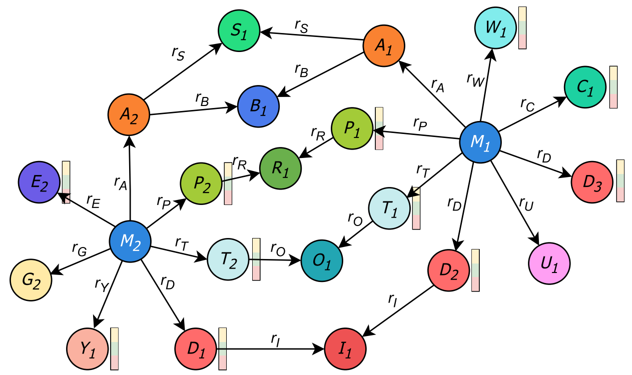

FIGURE 1. An example of a sub-graph from our knowledge graph. The circles and the rectangles surrounding entities denote entities and their attributes, respectively. Several entities, such as ***M***, ***A***, ***R***, ***O***, and ***D***, denote the Medical records, Animals, Drugs, Treatment code, and Disease, respectively.

Li et al. [38] have utilized the bilinear feature multiplication in a multi-model fusion to learn the text and image attributes with the entities in the KGs. However, most existing models learn representations through linear transformations, which could suffer from slow convergence and the capability to capture the heterogeneous property and graph structures in KGs.

Some studies have been proposed to automatically diagnose animal diseases through constructing an MKG and then learning the entities and relations [39], [40], [41]. For example, to diagnose dairy cows' diseases, Gao et al. [5] have constructed an MKG from EMRs and learned representations by using TransD method. Several studies adopt GAT to represent entities and relations in KGs and then combine them with the RNNs model to capture the patient's history for predicting disease [14], [16]. For example, Xu et al. [16] use GAT to learn representations through the combination with LSTMs as an auxiliary module for diagnosing pathology and disease [14]. However, learning KG structures independently with literal information may not enrich the learned representations and eventually reduce the model performance [42]. Unlike existing models, our model could learn different types of literal information combined with graph structures. Furthermore, the existing models learn different types of attributes with uniform weights between different entities and may not capture important "messages". In contrast, our model first fuses different types of entities and literal information through gate networks. Then, LiteralKG learns vector representations through coefficients between triplets in the KG, which could benefit from capturing graph structure using attentive weights.

III. MEDICAL KNOWLEDGE GRAPH CONSTRUCTION WITH ELECTRONIC MEDICAL RECORDS

We now represent our strategy to construct a medical knowledge graph from EMRs. A knowledge graph (KG) is a semantic network that represents heterogeneous data with different types of entities and relations in the real

TABLE 1. Summary of entities, relations, and attributes in our MKG. There are various types of relations between entities, including one-to-one, one-to-many, and many-to-many. Several entities, e.g., age, weight, and disease, carry the numerical or textual attributes described in the Attribute column.

Entity	#items	Notation	Description	Relation	Attribute
Medical Record	86,537	$\mathbb{M}$	The medical records of the visited companion animals that are connected to entities with various information about the animal, such as symptoms, age, weight, prescription, and the veterinarian's opinion of the companion animal who came for a checkup.	$r_A : \mathbb{M} \rightarrow \mathbb{A}$ $r_D : \mathbb{M} \rightarrow \mathbb{D}$ $r_Y : \mathbb{M} \rightarrow \mathbb{Y}$ $r_P : \mathbb{M} \rightarrow \mathbb{P}$ $r_T : \mathbb{M} \rightarrow \mathbb{T}$ $r_C : \mathbb{M} \rightarrow \mathbb{C}$ $r_E : \mathbb{M} \rightarrow \mathbb{E}$ $r_U : \mathbb{M} \rightarrow \mathbb{U}$ $r_W : \mathbb{M} \rightarrow \mathbb{W}$ $r_G : \mathbb{M} \rightarrow \mathbb{G}$	-
Animal	12,545	$\mathbb{A}$	It has a corresponding one-to-one number to the visited companion animal and is connected with the information with breed and species, and gender. e.g., "201-5010664"	$r_B : \mathbb{A} \rightarrow \mathbb{B}$ $r_S : \mathbb{A} \rightarrow \mathbb{S}$	-
Species	23	$\mathbb{S}$	The species of the visited companion animal. e.g., "Canine"	-	-
Breed	607	$\mathbb{B}$	The breed of the visited companion animal. e.g., "Poodle'"	-	-
Disease	133	$\mathbb{D}$	It means a disease in which a visited companion animal has or has been infected. The disease can be linked to the "Disease Category" entity.	$r_I : \mathbb{D} \rightarrow \mathbb{I}$	Textual attribute e.g., "liver tumor"
Symptom	86,537	$\mathbb{Y}$	The symptom of the visited companion animal at that time.	-	Textual attribute in the form of a sentence or word e.g., "Vomiting"
Drugs	1,563	$\mathbb{R}$	The prescribed drug for a companion animal. e.g., "ANB065"	-	-
Prescription	36,791	$\mathbb{P}$	The prescription such as the drug dose. This is more detailed than "Drugs" entity.	$r_R : \mathbb{P} \rightarrow \mathbb{R}$	Textual attribute e.g., "1 time 250mg, 3 times a day Amoxicillin/clavulanic acid Tab."
Treatment Code	5,952	$\mathbb{O}$	There are codes corresponding to treatment. e.g., "A022"	-	-
Treatment	146,113	$\mathbb{T}$	The companion animal can be received a treatment plan in the hospital. The treatment can be signed as a code. In this case, it is connected to "Treatment Code" entity	$r_O : \mathbb{T} \rightarrow \mathbb{O}$	Textual attribute e.g., "Intravenous injection"
Comment	132,926	$\mathbb{C}$	It is described by the veterinarian about the treatment response and disease progression.	-	Textual attribute e.g., "The animal breaths uncomfortably"
Age	24	$\mathbb{E}$	Natural numbers meaning the age of the visited companion animal are put in this entity's attribute.	-	Numerical attribute e.g., 14
Age Group	4	$\mathbb{U}$	There are four types of age groups including "Infancy"(age $<$ 1), "Adult" ($1 \leq$ age < 7), "Old-age" ($7 \leq$ age < 13), "Super-aged" ($13 \leq$ age)	-	-
Gender	4	$\mathbb{G}$	There are only four genders of companion animals in our MKG, including unspayed female, spayed female, unneutered male, and neutered male.	-	-
Weight	85,397	$\mathbb{W}$	The weight of the visited companion animal.	-	Numerical attribute e.g., 2.5
Disease Category	16	$\mathbb{I}$	It is the category of disease. Different diseases can be connected to the same disease category. e.g., "nervous system"	-	-

world [30], [31]. Formally, a KG is a set of triples where each triple is formed of $\langle h, r, t \rangle$, where h, r, t refers to the head, relation, and tail, respectively [43]. Medical knowledge graph (MKG) is a knowledge graph that represents the relations in the healthcare area through representing medical data, i.e., electronic medical records (EMRs) [4]. These records contain various types of information, such as patients, diseases, medicines, and symptoms [4], [44].

We now explain our proposed strategy to construct a KG and entity relations from EMRs. In our study, we constructed a KG, which is composed of 85,965 EMRs from 31 companion animal hospitals, collected from IntoCNS company.[1] In EMRs, a record is a collection of medical properties, such as companion animal, symptoms, disease, and the veterinarian's decisions for each companion animal visit. We generate entities from the medical properties. There are a total of sixteen entity types and fifteen relation types in our KG. Table 1 shows the detailed statistics of entity types and their relations. In each record, the types and names of the entities are extracted from the fields and elements in EMRs, respectively. As the symptoms ($\mathbb{Y}$) are recorded in the form of textual sentences, they could be represented

[1] http://intoh.monoalliance.com/en/

under the different views of veterinarians. It shows that when we construct our KG, each textual symptom may contain different textual sentences, even if they could have several similar symptoms. We then apply Fasttext [43] to mapping textual information into fixed-length vectors to generate the initial symptom features. If textual sentences have some common words, such as "fever" and "vomiting", the Fasttext could learn their contextual information and semantics between them and then map them to be close in the latent space. As a result, we constructed 86,537 symptom vectors, which could be different from each other. It is worth noting that the relationship between symptoms ($\mathbb{Y}$) and medical records ($\mathbb{M}$) is a one-to-one relationship due to the text embedding.

Figure 1 illustrates simplified entities and their relations in our MKG. Our purpose is first to construct entities and then build different types of relations between entities in KGs from EMRs. Let $\mathbb{M}$ denote the set of medical record entities, and $\mathbb{A}$ refers to the animal entity set. We construct one-to-many relations between animal entity $\mathbb{A}$ and $\mathbb{M}$ since an animal A_i could be examined many times and have more than one medical record. Other entities in our MKG are constructed into a structure that satisfies their natural types of relations. For example, the relations between $\mathbb{M}$ and $\mathbb{D}$ are many-to-many since one animal could suffer many diseases or many animals have the same diseases. According to the hospital process, when a companion animal visits a hospital, a veterinarian could make a medical record ($\mathbb{M}$) containing the symptoms ($\mathbb{Y}$) and then predict the diseases ($\mathbb{D}$). Therefore, the symptom ($\mathbb{Y}$) and disease ($\mathbb{D}$) are connected through the medical records ($\mathbb{M}$), which aligns with the nature of the relationship. To predict the disease ($\mathbb{D}$) of an animal ($\mathbb{A}$), the veterinarian considers various additional factors, such as the species of the animal ($\mathbb{S}$), Breed ($\mathbb{B}$), previous medical records, and so on. After considering all the related information, the veterinarian could predict the animal diseases as represented in our knowledge graph. Therefore, the current symptom is one of the factors that a veterinarian could consider to give a disease prediction. By representing such relations in the knowledge graph, our model could learn the representations and capture the relationship between various entities, such as symptom ($\mathbb{Y}$) and disease ($\mathbb{D}$).

Note that the entities are composed of different types of attributes, such as numeric and text attributes. For example, the disease, symptom, treatment, age, and weight entities have textual or numerical attributes that can be divided into numeric or text groups. For textual attributes, we then encode the attributes by using Fasttext [45] to capture the semantic contexts of the textual attributes. Otherwise, if an entity does not contain an attribute, its attribute embedding should be known as a non-attribute entity. In this case, the textual or numerical attribute vectors will be presented as vectors of zero. Accordingly, we then fuse entity and attributed features through gate networks to generate literal enriched embedding vectors for entities.

IV. LITERAL-AWARE MEDICAL KNOWLEDGE GRAPH REPRESENTATION LEARNING

In this section, we first represent how to fuse entities and different attribute types into unified representations. Then, we will introduce the architecture of our model in detail. In addition, we represent a pre-training task that could learn the representations in the pretext tasks and then apply them to downstream tasks, i.e., predicting diseases.

A. FUSING ENTITY AND ATTRIBUTE FEATURES

As mentioned earlier, the entities contain two main types of attributes, including numerical and textual attributes. We first design a fusion layer that contains a gate function to transform different types of attributes into unified vectors. The textual attributes, such as disease, symptom, prescription, treatment, and comment, are transformed into vectors through Fasttext [45]. Numerical attributes, such as age and weight entities, are normalized and transformed directly into feature vectors [21]. In several cases, missing values are represented by '-1' as missing features. We then transform entities into the shared space to generate the unified literal-enriched vectors by using a gate function [21]. Formally, the output representations for an i-th entity can be defined as:

$$\begin{aligned} h_i^{(0)} &= g(e_i, n_i, t_i) = \mu \odot \nu + (1-\mu) \odot \hat{e}_i \\ \text{where, } \mu &= \sigma_1\left(\hat{e}_i + \hat{n}_i + \hat{t}_i + b\right), \\ \nu &= \sigma_2\left(W \cdot (e_i \| n_i \| t_i)\right), \end{aligned} \tag{1}$$

and $\odot$ is the Hadamard product, σ_1 and σ_2 are sigmoid and tanh activation functions, respectively. The entity vector $e_i \in \mathbb{R}^E$, the numerical attribute vector $n_i \in \mathbb{R}^N$ and the textual attribute vector $t_i \in \mathbb{R}^T$ are combined and transformed into a vector $h_i^{(0)} \in \mathbb{R}^h$ with fixed dimension h by a gate function g. e_i, n_i, t_i are passed through a linear projection $\hat{e}_i = W_E \cdot e_i$, $\hat{n}_i = W_N \cdot n_i$, and $\hat{t}_i = W_T \cdot t_i$, respectively. $W_E \in \mathbb{R}^{E\times h}$, $W_N \in \mathbb{R}^{N\times h}$, $W_T \in \mathbb{R}^{T\times h}$ and $b \in \mathbb{R}^h$ and $W \in \mathbb{R}^{(E+N+T)\times h}$ are learnable parameters.

B. LEARNING GLOBAL AND LOCAL STRUCTURAL FEATURES

After generating literal-enriched embedding vectors, we use an attention mechanism to learn the co-coefficients across triplets. Figure 2 shows the overall architecture of our model. The literal-enriched embedding vectors will be passed through the attentive embedding propagation layers. Formally, the representation of a entity e_i at l-th layer could be updated as:

$$h_i^{(l)} = f_*\left(h_i^{(l-1)}, h_{\mathcal{N}_i}^{(l-1)}\right) \tag{2}$$

$$h_{\mathcal{N}_i}^{(l-1)} = \sum_{\forall (h_i, r_j, t_k)\in \mathcal{N}_i} \pi(h_i, r_j, t_k) t_k^{(l-1)} \tag{3}$$

where $h_i^{(l)}$ is the output representation of an entity e_i at l-th layer, $\mathcal{N}_i$ denotes a set of neighbours of entity e_i which is composed with the triple (h_i, r_j, t_k) if and only if there is a link between h_i and t_k, and f_* is a GNN aggregator.

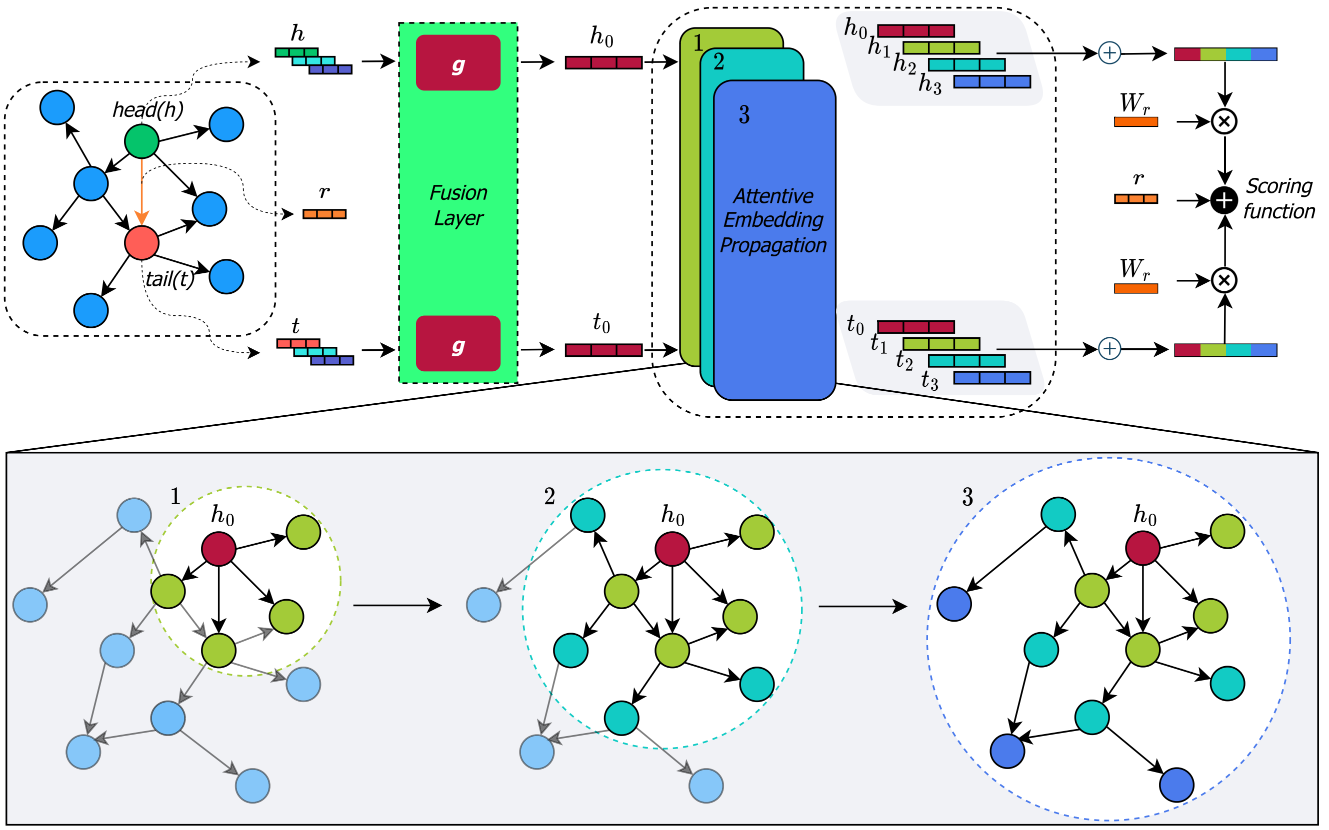

FIGURE 2. **The overall architecture of LiteralKG. The entities and attributes are fused into unified vectors through a gate function $g(\cdot)$. The unified vectors are then passed through attentive embedding propagation layers to generate the output representations. The output representations are then concatenated with the initial embeddings to generate final representations by optimizing a score function.**

Since neighbours with different relation types could contribute differently to a target entity, we aim to aggregate features of neighbouring nodes with attentional weights to benefit the model in learning different relations between entities. The attentive weights could, therefore, describe the nature of the relations between entities in KGs. More formally, the attentive scores are computed as follows:

$$\pi(h, r, t) = (W \cdot t)^{\intercal} \tanh(W \cdot h + r) \tag{4}$$

$$\pi(h, r, t) := \frac{exp(\pi(h, r, t))}{\sum_{(h,r',t') \in \mathcal{N}_i} exp(\pi(h, r', t'))}. \tag{5}$$

Since different GNN aggregators show individual characteristics, they could benefit in different ways to explore the explicit and complex KG structures. Therefore, we aim to investigate which types of GNN aggregators could contribute to the overall performance of our model. In this study, we use four aggregators, including GCN, GraphSAGE, Bi-Interaction, and GIN aggregators. First, GCN aggregator [46] combines the entity feature and its neighbour's features by a sum operator. At each k-th layer, the model aggregates k-hop neighbourhood features to leverage the graph structure and generate output representation. A non-linearity transformation is then utilized to transform the output features before updating the representations. The GCN aggregator is defined as:

$$f_{GCN}^{(l)}\left(h_i^{(l-1)}, h_{\mathcal{N}_i}^{(l-1)}\right) = \sigma\left(W^{(l)}\left(h_i^{(l-1)} + h_{\mathcal{N}_i}^{(l)}\right)\right) \tag{6}$$

where σ denotes the activation function, $h_i^{(l-1)}$ and $h_{\mathcal{N}_i}^{(l-1)}$ are the output of the previous layer and the aggregated neighbourhood features of an entity, respectively, and $W^{(l)}$ is the learnable transformation matrix at l-th layer.

GraphSAGE aggregator replaces sum by concatenation operator to distinguish between the entity feature and its neighbourhood aggregation feature [17], [46]. Note that the difference between GCN and GraphSAGE is that GraphSAGE learns the topological structure for each target node neighbourhood through random walks, which could then calculate node embeddings in an inductive manner. GraphSAGE aggregator is described as follows:

$$f_{SAGE}^{(l)}\left(h_i^{(l-1)}, h_{\mathcal{N}_i}^{(l-1)}\right) = \sigma\left(W^{(l)}\left(h_i^{(l-1)} \| h_{\mathcal{N}_i}^{(l-1)}\right)\right) \tag{7}$$

where $\|$ is the concatenation.

Wang et al. [46] have proposed a Bi-Interaction aggregator which combines the GCN-based strategy and the element-wise product of the target node and its neighbour feature. The element-wise product could assist the model in learning the similarity between target nodes and neighbourhoods. Formally, the Bi-Interaction equation is considered as follows:

$$f_{BI}^{(l)}\left(h_i^{(l-1)}, h_{\mathcal{N}_i}^{(l-1)}\right) = \sigma\left(W_1^{(l)}\left(h_i^{(l-1)} + h_{\mathcal{N}_i}^{(l-1)}\right)\right) + \sigma\left(W_2^{(l)}\left(h_i^{(l-1)} \odot h_{\mathcal{N}_i}^{(l-1)}\right)\right), \tag{8}$$

where $W_1^{(l)}$ and $W_2^{(l)}$ are the learnable parameters.

Inspired by 1-d Weisfeiler-Lehman (WL) isomorphism testing, GIN [18] aims to maximize the GNNs power up to 1d-WL test. The key difference between GIN and other aggregators is that GIN could map different sub-structures into different representations, leading to the power to distinguish non-isomorphic sub-structures. At each layer, the GIN aggregator updates the representations through the entity and neighbour features with a sum aggregator. Note that GIN does not include any normalization during updating node features. The GIN aggregator can be described as:

$$f_{GIN}^{(l)}\left(h_i^{(l-1)}, h_{\mathcal{N}_i}^{(l-1)}\right) = \sigma\left[FC\left(\left(1+\epsilon^{(l)}\right) \cdot h_i^{(l-1)} + h_{\mathcal{N}_i}^{(l-1)}\right)\right], \tag{9}$$

where ϵ can be a fixed scale or learnable parameter, and FC denotes the fully connected layer. In the GIN model, the embeddings of a node i at layer l-th could be updated iteratively with $h_i^{(l)} = \phi\left((1+\varepsilon^{(l-1)}) \cdot h_i^{(l-1)} + f\left(h_{\mathcal{N}_i}^{(l-1)}\right)\right)$, where $f(\cdot)$ operates on multisets and $\phi(\cdot)$ denotes injective. We then model $f^{(l)} \circ \phi^{(l)}$ with one fully connected layer as shown in Equation 9, following the work [18].

Stacking more GNN layers can lead to the over-smoothing problem, and eventually lead to reducing the performance of the model. Therefore, we add initial residual connections and identity mapping, following the work [22]. Formally, the formula of residual connection and identity mapping can be described as:

$$\begin{aligned} H^{(l+1)} &= \sigma\left[\left((1-\alpha_l)\tilde{P}H_i^{(l)} + \alpha_l H_i^{(0)}\right) \times \left((1-\beta_l)I_n + \beta_l W^{(l)}\right)\right] \\ \text{where } \beta_l &= \log\frac{\lambda}{1+l} \\ \tilde{P} &= (D+I_n)^{-1/2}(A+I_n)(D+I_n)^{-1/2}, \end{aligned} \tag{10}$$

and $\tilde{P}$ is a graph convolution matrix, λ and α_l are hyper-parameters.

To compute the final representation of an i-th entity e_i, we first concatenate all the output representations of GNN layers. As the local and global graph structures are important to represent entities, we aim to combine the initial feature with the output features of all GNN layers. Therefore, the representations could learn the local and global graph structures [11], [46]. We then apply a linear function followed by an activation function to transform the entity vectors into final representations:

$$e_i = \sigma\left(W \cdot \Big\|_{k=1}^{K}\left(h_i^{(k)}\right) + b\right) \tag{11}$$

where K presents the number of GNN layers, b is the bias of the linear function, and $W \in \mathbb{R}^{h\times K} \rightarrow \mathbb{R}^d$ is the weight matrix.

C. PRE-TRAINING WITH FOCUSING ON MULTI-RELATIONAL STRUCTURES

We now represent our pre-training task to learn multi-relational structures between different types of entities in KG. In this study, we aim to preserve the co-coefficients across all triplets through a scoring function. For each triplet, entity embedding vectors are first transformed into a shared space through a projection matrix. Then, we use a triplet score function to calculate their relation score [31]. To preserve all the entity relations, we aim to maximize all the positive triplets coming from KGs and minimize all the negative triplets that are not coming from KGs [30], [31]. Formally, the scoring function is defined as:

$$f_{score} = \hat{y}(h, r, t) = W_r h + r - W_r t \tag{12}$$

where h, t, and r are the output representations of head, tail, and relation, and W_r is a projection matrix to map entities and relations onto the r space. A triplet loss function is computed to compare the positive and negative triple pairs defined as:

$$\mathcal{L}_{\mathcal{P}}(\mathcal{T}) = \sum_{\forall(h,r,t)\in\mathcal{T}} -\ln\sigma\left(\hat{y}(h,r,\bar{t}) - \hat{y}(h,r,t)\right) + \lambda||\Theta||_2^2, \tag{13}$$

where $\mathcal{T}$ denotes the set of triplets, $(h, r, \bar{t})$ denote the negative triplet, Θ refers to L_2 regularization parameter.

D. FINE-TUNING FOR ANIMAL DISEASE DIAGNOSTICS

We now apply learned LiteralKG and representations to new downstream task, such as predicting animal diseases. We first compute a coefficient score to measure the relationship between each head and tail pair. The equation for calculating the coefficient score can be formulated as follows:

$$\hat{y}_{h,t} = \hat{y}(h, t) = \langle\phi(h), \phi(t)\rangle = FC(W_r h || W_r t), \tag{14}$$

where h and t denote the learned representations of head and tail, respectively, $||$ is the concatenation operator, W_r is the transformation matrix corresponding to the relation between h and t, FC denotes the fully connected layers to get the prediction output, and $\phi(\cdot)$ represents the final embeddings computed from the Equation 11.

For disease diagnosis, the model classifies the coefficient score between two entities into binary digits. With pre-trained embeddings, two observed entity vectors are first projected into the relation space. Their embedding vectors are then concatenated and transformed into a one-dimensional class by MLP layers to form the coefficient score. It is used to identify the relationship probability among the observed entities. Additionally, a binary cross entropy loss function is utilized for training classification. The loss function is described as:

$$\mathcal{L}_{\mathcal{F}}(\mathcal{M}) = -\sum_{\forall(M_k,D_i)\in\mathcal{M}} \left[y_{M_k,D_i} \log\left(\hat{y}_{M_k,D_i}\right) + (1-y_{M_k,D_i}) \log\left((1-\hat{y}_{M_k,D_i})\right)\right] \tag{15}$$

where M_k and D_i are the medical record and disease entity, y_{M_k,D_i} and $\hat{y}_{M_k,D_i}$ are the actual and predicted outputs, and $\mathcal{M}$ is the collection of all the training medical records containing positive and negative disease information. It is worth noting that each medical record is connected with various types of entities, such as species of the animal ($\mathbb{S}$), Breed ($\mathbb{B}$), and previous medical records, which could benefit our model in learning the additional information to predict the disease ($\mathbb{D}$). As the relation between $\mathbb{M}$ and $\mathbb{D}$ is a many-to-many connection, our model could discover the relations between medical records ($\mathbb{M}$) and disease entities ($\mathbb{D}$) and then map them to be close in the latent space if they are connected in our knowledge graph. In addition, since the relation between symptoms ($\mathbb{Y}$) and disease ($\mathbb{D}$) is a one-to-one connection and through medical record ($\mathbb{M}$), our model could also capture the similarity between them by the GNN aggregation mechanism. It indicates that if there is any relation between symptoms ($\mathbb{Y}$), medical records ($\mathbb{M}$), and disease ($\mathbb{D}$), our model could learn the similarity between them by aggregating the neighbourhood information iteratively with several GNN layers.

V. EXPERIMENTS

In this section, we provide extensive experimental results to validate the performance of our model versus baselines. In addition, we conduct ablation studies to investigate the contribution of the combination of different types of relations as well as residual connection and identity mapping to the overall performance.

A. EXPERIMENTAL SETTINGS

1) DATASET DESCRIPTION

As mentioned earlier, we collected 85,965 EMRs from 31 animal hospitals and transformed to a knowledge graph. After transforming, our MKG contains a total of 595,172 entities and 16 relation types. Note that we sampled three negative triplets for each positive triplet in the experiments. We first pre-trained our model in a self-supervised manner without using any label information. Then, we fine-tuned the LiteralKG model to learn the knowledge for link prediction tasks. For the fine-tuning task, we conducted the experiment by randomly sampling training, validation, and testing sets of size 60%, 20%, and 20%, respectively.

2) EVALUATION METRICS

Since our task is a binary classification problem, we utilized several evaluation metrics, including accuracy (Acc), precision (P), recall (R), and F_1. The evaluation metrics are defined as follows:

$$Acc = \frac{|D^+_{true} \cup D^-_{true}|}{|D^+ \cup D^-|} \quad P = \frac{|D^+_{true}|}{|D^+|}$$
$$R = \frac{|D^+_{true}|}{|D^+_{true} \cup D^-_{false}|} \quad F_1 = 2\frac{(PR)}{(P+R)} \tag{16}$$

where the D^+_{true} and D^-_{true} denote the correct predictions for positive and negative diseases, respectively, and D^-_{false} is the incorrect predictions for negative diseases. These criteria are intended to assess how well our model performs compared to baselines.

3) BASELINES

We compare our model to relevant translation-based methods and GNN models, which have gained remarkable success in KG representation learning. Translation-based models learn representations by mapping the entities and their relations into latent space through translations.

- **TransE** [30]. The model uses a simple scoring function to represent the similarity between pairs of entities and map them into latent space through a linear translation. Note that all types of entities and relations are represented in the same space.
- **TransR** [31]. TransR uses projection matrices to map various types of entities into different relation spaces and then construct translations between entities.
- **SMR** [4]. The idea of SMR is to use TransR to transform entities into latent space linearly. Meanwhile, SMR uses the LINE [47] model to capture the similarity between entities, which successfully addresses homogeneous graphs.

Furthermore, we also compare our model with recent GNNs, which could learn high-order sub-structures and semantic relations in KGs. There are three GNN baselines, including KGNN, KGNMDA, and LaGAT model, as:

- **KGNN** [10] learn KG structures through a local receptive to aggregate neighbour features and topological information. KGNN could also capture the high-order structures surrounding target entities and semantic relations to learn global structural information.
- **KGNMDA** [48] represent the relations between microbes and diseases based on the Gaussian kernel and then learn the similarity between them in an uncertain manner. They then use a linear transformation to predict the scores across microbe-disease relations.
- **LaGAT** [11] extends KGNN by using an attentive mechanism, which could learn different weighted messages contributed from different entities. Furthermore, the outputs of different attentive embedding propagation layers are concatenated and then contribute to the final representations.

4) IMPLEMENTATION DETAILS

Our model is implemented based on the Pytorch library. All the experiments were done in two GPU servers with four

TABLE 2. The performance of LiteralKG and baselines in terms of accuracy, recall, precision and F_1. The top two are highlighted by first and second.

Model	Accuracy	Recall	Precision	F_1
TransE [30]	0.5937	0.5617	0.6001	0.5802
TransR [31]	0.5732	0.5915	0.5706	0.5808
SMR [4]	0.5872	0.5669	0.5909	0.5787
KGNN [10]	0.7890	0.7021	0.8499	0.7689
KGNMDA [48]	0.8130	0.7111	0.8932	0.7918
LaGAT [11]	0.8545	0.7988	0.8988	0.8459
LiteralKG	0.8616	0.9357	0.8150	0.8712

NVIDIA RTX A5000 GPUs for each (24GB RAM/GPU). Adam optimizer [49] was applied to the pre-training and fine-tuning phases. We applied Leaky ReLU as an activation function in the aggregation layers. The range of learning rate is $\in \{0.0001, 0.00005, 0.00001\}$. The hidden dimension of the GNN layers $\in \{16, 32, 64, 128\}$. The number of GNN layers $\in \{2, 4, 8, 16\}$. The dropout ratio $\in \{0.1, 0.5\}$ after each layer. For fair comparisons with the baselines, the hyper-parameters in all baselines were tuned in the same range, including the learning rate $\in \{0.001, 0.0001\}$, hidden dimension $\in \{16, 32, 64, 128\}$ and the number of layers $\in \{1, 2, 4\}$. The source code is available at https://github.com/NSLab-CUK/LiteralKG.

B. PERFORMANCE ANALYSIS

Table 2 shows the performance of our model and baselines in terms of accuracy, recall, precision, and F_1. We have the following observations: **(1)** LiteralKG with pre-training outperformed baselines in most measurements. Specifically, our proposed model reached the best value at 0.9357 regarding Recall measurement. We argue that our model could learn the literal information well to maximize the relations between entities. **(2)** Translation-based models, i.e., TransE, TransR, and SMR, showed low performance in predicting disease. We argue that these models overlooked literal information and eventually could not capture the complex relations since they only learn representations through simple linear transformations. **(3)** LaGAT showed competitive performance compared to our model. We argue that as LaGAT could learn graph structure through the attention mechanism, the model thus could learn attentive weights contributed from different neighbourhood entities in KGs.

C. ABLATION STUDIES

1) ON THE IMPORTANCE OF LEARNING LITERAL INFORMATION

Table 3 shows how the literal information contributes to the overall performance of our model. We have the following observations: **(1)** LiteralKG showed the best performance with the combination of numerical and textual information in most measurements. We argue that both types of literal information could benefit our model to explore similar entities in the KG and thus contribute to the overall performance.

TABLE 3. The results for evaluating the efficiency of leveraging literals. The first column contains the combining literal strategies. There are four cases, including without (w/o) literal, without textual, without numerical, and the combination of numerical and textual features. The top two are highlighted by first and second.

Combining literals	Accuracy	Recall	Precision	F_1
w/o literal	0.8223	0.8664	0.7962	0.8298
w/o textual	0.8424	0.8544	0.8344	0.8443
w/o numerical	0.8529	0.9116	0.8158	0.8610
numerical + textual	0.8616	0.9357	0.8150	0.8712

TABLE 4. Link prediction performance on the Residual Connection and Identity Mapping (R.C& I.M) in GNN aggregators. × and ○ denote a setting ignoring and adopting R.C& I.M, respectively. The top two are highlighted by first and second.

R.C& I.M	Aggregator	Accuracy	Recall	Precision	F_1
×	GCN	0.8532	0.9419	0.8000	0.8652
	GraphSAGE	0.8580	0.9257	0.8154	0.8670
	Bi-Interaction	0.8545	0.9123	0.8178	0.8625
	GIN	0.8387	0.8621	0.8236	0.8424
○	GCN	0.8527	0.9170	0.8125	0.8616
	GraphSAGE	0.8300	0.8882	0.7956	0.8394
	Bi-Interaction	0.8616	0.9357	0.8150	0.8712
	GIN	0.8490	0.9372	0.7967	0.8612

TABLE 5. The performance of LiteralKG on the role of the pre-training task. × and ○ denote a setting without a pre-trained model and utilizing the pre-trained model, respectively. The top two are highlighted by first and second.

Pre-training	Aggregator	Accuracy	Recall	Precision	F_1
×	GCN	0.8183	0.7371	0.8801	0.8022
	GraphSAGE	0.8109	0.7429	0.8597	0.7971
	Bi-Interaction	0.7976	0.7083	0.8622	0.7777
	GIN	0.8013	0.8053	0.7989	0.8021
○	GCN	0.8527	0.9170	0.8125	0.8616
	GraphSAGE	0.8300	0.8882	0.7956	0.8394
	Bi-Interaction	0.8616	0.9357	0.8150	0.8712
	GIN	0.8490	0.9372	0.7967	0.8612

(2) We find out that the performance of our model increased slightly in comparison with without using numerical information, from 0.8529 to 0.8616 and from 0.8610 to 0.8712 in terms of accuracy and F_1, respectively. It indicates that the textual information contributes considerably to the overall performance compared to numerical attributes. Thanks to the textual features, LiteralKG performs well even without numerical features. **(3)** When combining different types of attributes, the recall measurement increased from 0.8664 to 0.9357, which shows a good point for maximizing the ability to diagnose patients with a correct disease. It indicates that the gating mechanism transformed the attributes and entity embedding vectors into the shared space and eventually contributed to the overall performance.

2) ON THE IMPORTANCE OF RESIDUAL CONNECTION AND IDENTITY MAPPING

We further conducted experiments to evaluate the effectiveness of residual connection and identity mapping shown in Table 4. We have the following observations: **(1)** The overall

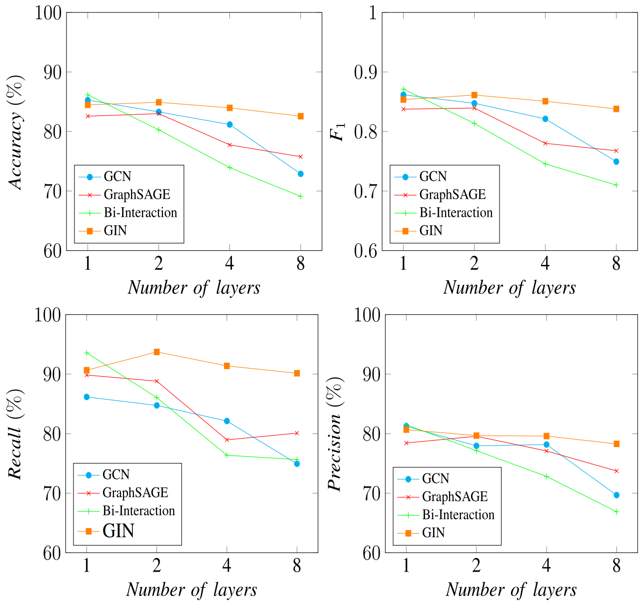

FIGURE 3. Performance on link prediction over attentive embedding propagation layers on four types of aggregators, including GCN, GraphSAGE, Bi-Interaction, and GIN aggregators.

performance increased slightly by applying the residual connection and identity mapping. It indicates that even though KGs have complex relations and heterophily properties, residual connections and identity mapping could act as auxiliary modules to contribute to the overall performance of LiteralKG. In other words, the two modules could help LiteralKG prevent over-smoothing problem and improve the performance of our model. **(2)** In comparison with GraphSAGE, the performance of our model with residual connection was reduced from 0.8580 to 0.8300 and from 0.8670 to 0.8394 in terms of accuracy and F_1, respectively. It indicates that the residual connection does not contribute much to our model performance as GraphSAGE sampled the neighbourhoods based on random walks. **(3)** For the GIN aggregator, using residual connection can improve the model performance from 0.8174 to 0.8378 in terms of F_1 measure. Note that GIN aggregators aim to map different sub-structures into different representations, leading to the power of 1d-WL isomorphism testing. This implies that residual connections could contribute to the model performance as GIN aggregators could ignore the original features of entities.

3) ON THE PRE-TRAINING PHASE

Table 5 shows the link prediction performance on the pre-training phase evaluation. We discovered that our pre-trained model could show comparable performance in most of the aggregators. In particular, LiteralKG reached the highest values in the Bi-Interaction aggregator at 0.8616, 0.8150, and 0.8712 in terms of accuracy, precision, and F_1, respectively. We argue that the triplet scoring function in the pre-training phase first learns to maximize relation proximity across triplets in the KG. Therefore, it could learn and capture the entity similarity, which can contribute to the performance of the fine-tuning task in link prediction, in which the link is also considered by representation proximity. In GIN, applying the pre-trained model improves performance on F_1 and accuracy from 0.8021 to 0.8612 and 0.8013 to 0.8490, respectively. These results, therefore, explain the efficiency of the pre-trained model, which satisfies the RQ2.

D. SENSITIVITY ANALYSIS

Figure 3 shows the performance of our model by applying different range of GNN layers. We have the following observations: **(1)** This result illustrates that LiteralKG with a 1-layer or 2-layer can efficiently learn the KG structure in most of the aggregators. For the Bi-Interaction aggregator, LiteralKG achieves the best performance when using only one layer, which captures 1-hop neighbourhood entities. We argue that as Bi-Interaction could capture the graph structure through sum aggregators and element- wise product, the aggregators could suffer the over-smoothing problem when staking more layers. In other words, Bi-Interaction aggregators could not distinguish sub-structures as the aggregators learned the similarity between target nodes and neighbours at the first layers. **(2)** It is worth noting that GIN aggregators showed the highest performance when staking more GNN layers compared to other aggregators. We argue that as the power of GIN achieves nearly 1d-WL isomorphism testing, the model then could handle the over-smoothing problem even staking more GNN layers. In other words, GIN aggregators could distinguish different sub-structures even staking more layers.

VI. CONCLUSION AND FUTURE WORK

In this study, we propose a knowledge graph embedding model, LiteralKG, which could learn different types of entity attributes to diagnose companion animal disease. By doing so, we first constructed an MKG from various EMRs collected from 31 animal hospitals. Then, LiteralKG fuses different types of literals into unified representations through a gating network. We then use the attention mechanism with the initial feature to learn the coefficients across triplets to capture local and global graph structures. The experiment results show that our model outperforms the shallow KG embedding and GNN-based models due to the improvements from leveraging the literal features and the efficiency of the pre-trained phase. Furthermore, we present a pre-training task that could learn graph structure and its properties without using any label information to generate learned representations. The pre-trained model with representations could then be used for prediction tasks. Besides, the negative samples from our data are sampled randomly so that it can affect the overall performance of LiteralKG. In future work, we plan to build useful sampling strategies to effectively build positive and negative samples, such as graph clustering and sub-graph sampling methods.

ACKNOWLEDGMENT

(Van Thuy Hoang and Thanh Sang Nguyen are co-first authors.)

REFERENCES

[1] Y. Zhang, A. Nie, A. Zehnder, R. L. Page, and J. Zou, "VetTag: Improving automated veterinary diagnosis coding via large-scale language modeling," *NPJ Digit. Med.*, vol. 2, no. 1, p. 35, May 2019, doi: 10.1038/s41746-019-0113-1.

[2] M. R. Boland, M. L. Casal, M. S. Kraus, and A. R. Gelzer, "Applied veterinary informatics: Development of a semantic and domain-specific method to construct a canine data repository," *Sci. Rep.*, vol. 9, no. 1, pp. 1–13, Dec. 2019.

[3] S. V. Inácio, J. F. Gomes, A. X. Falcão, C. T. N. Suzuki, W. B. Nagata, S. H. N. Loiola, B. M. dos Santos, F. A. Soares, S. L. Rosa, C. B. Baptista, G. B. Alves, and K. D. S. Bresciani, "Automated diagnosis of canine gastrointestinal parasites using image analysis," *Pathogens*, vol. 9, no. 2, p. 139, Feb. 2020.

[4] F. Gong, M. Wang, H. Wang, S. Wang, and M. Liu, "SMR: Medical knowledge graph embedding for safe medicine recommendation," *Big Data Res.*, vol. 23, Feb. 2021, Art. no. 100174, doi: 10.1016/j.bdr.2020.100174.

[5] M. Gao, H. Wang, W. Shen, Z. Su, H. Liu, Y. Yin, Y. Zhang, and Y. Zhang, "Disease diagnosis of dairy cow by deep learning based on knowledge graph and transfer learning," *Int. J. Bioautomation*, vol. 25, no. 1, pp. 5–12, 2021.

[6] A. N. Paynter, M. D. Dunbar, K. E. Creevy, and A. Ruple, "Veterinary big data: When data goes to the dogs," *Animals*, vol. 11, no. 7, p. 1872, Jun. 2021.

[7] J. Li, J. Gao, B. Feng, and Y. Jing, "PlagueKD: A knowledge graph–based plague knowledge database," *Database*, vol. 2022, Nov. 2022, Art. no. baac100.

[8] L. Murali, G. Gopakumar, D. M. Viswanathan, and P. Nedungadi, "Towards electronic health record-based medical knowledge graph construction, completion, and applications: A literature study," *J. Biomed. Informat.*, vol. 143, Jul. 2023, Art. no. 104403.

[9] L. Li, P. Wang, Y. Wang, S. Wang, J. Yan, J. Jiang, B. Tang, C. Wang, and Y. Liu, "A method to learn embedding of a probabilistic medical knowledge graph: Algorithm development," *JMIR Med. Informat.*, vol. 8, no. 5, May 2020, Art. no. e17645.

[10] X. Lin, Z. Quan, Z.-J. Wang, T. Ma, and X. Zeng, "KGNN: Knowledge graph neural network for drug-drug interaction prediction," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Yokohama, Japan, vol. 380, Jul. 2020, pp. 2739–2745, doi: 10.24963/ijcai.2020/380.

[11] Y. Hong, P. Luo, S. Jin, and X. Liu, "LaGAT: Link-aware graph attention network for drug–drug interaction prediction," *Bioinformatics*, vol. 38, no. 24, pp. 5406–5412, Dec. 2022, doi: 10.1093/bioinformatics/btac682.

[12] C. Nusai, S. Cheechang, S. Chaiphech, and G. Thanimkan, "Swine-vet: A web-based expert system of swine disease diagnosis," *Proc. Comput. Sci.*, vol. 63, pp. 366–375, 2015.

[13] Suharjito, Diana, Yulyanto, and A. Nugroho, "Mobile expert system using fuzzy Tsukamoto for diagnosing cattle disease," *Proc. Comput. Sci.*, vol. 116, pp. 27–36, 2017.

[14] X. Chai, "Diagnosis method of thyroid disease combining knowledge graph and deep learning," *IEEE Access*, vol. 8, pp. 149787–149795, 2020, doi: 10.1109/ACCESS.2020.3016676.

[15] W. Lan, Y. Dong, Q. Chen, R. Zheng, J. Liu, Y. Pan, and Y.-P.-P. Chen, "KGANCDA: Predicting circRNA-disease associations based on knowledge graph attention network," *Briefings Bioinf.*, vol. 23, no. 1, Jan. 2022, Art. no. bbab494, doi: 10.1093/bib/bbab494.

[16] X. Xu, X. Xu, Y. Sun, X. Liu, X. Li, G. Xie, and F. Wang, "Predictive modeling of clinical events with mutual enhancement between longitudinal patient records and medical knowledge graph," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Auckland, New Zealand, Dec. 2021, pp. 777–786, doi: 10.1109/ICDM51629.2021.00089.

[17] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 30th Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 1024–1034. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

[18] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*. New Orleans, LA, USA, May 2019, pp. 1–17. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km

[19] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," 2015, *arXiv:1511.03677*.

[20] H. M. Sergieh, T. Botschen, I. Gurevych, and S. Roth, "A multimodal translation-based approach for knowledge graph representation learning," in *Proc. 7th Joint Conf. Lexical Comput. Semantics (SEM@NAACL-HLT)*. New Orleans, LA, USA: ACL, Jun. 2018, pp. 225–234, doi: 10.18653/v1/s18-2027.

[21] A. Kristiadi, M. A. Khan, D. Lukovnikov, J. Lehmann, and A. Fischer, "Incorporating literals into knowledge graph embeddings," in *Proc. 18th Int. Semantic Web Conf. (ISWC)*, in Lecture Notes in Computer Science, vol. 11778. Auckland, New Zealand: Springer, Oct. 2019, pp. 347–363, doi: 10.1007/978-3-030-30793-6_20.

[22] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, vol. 119, Jul. 2020, pp. 1725–1735. [Online]. Available: http://proceedings.mlr.press/v119/chen20v.html

[23] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[24] H.-J. Jeon, G.-S. Choi, S.-Y. Cho, H. Lee, H. Y. Ko, J. J. Jung, O.-J. Lee, and M.-Y. Yi, "Learning contextual representations of citations via graph transformer," in *Proc. 2nd Int. Conf. Hum.-Centered Artif. Intell. (Computing4Human)*, Da Nang, Vietnam, Oct. 2021, pp. 1–9.

[25] H.-J. Jeon, M.-W. Choi, and O.-J. Lee, "Day-ahead hourly solar irradiance forecasting based on multi-attributed spatio-temporal graph convolutional network," *Sensors*, vol. 22, no. 19, p. 7179, Sep. 2022.

[26] J. Zhang, H. Zhang, C. Xia, and L. Sun, "Graph-bert: Only attention is needed for learning graph representations," 2020, *arXiv:2001.05140*.

[27] V. T. Hoang, H.-J. Jeon, E.-S. You, Y. Yoon, S. Jung, and O.-J. Lee, "Graph representation learning and its applications: A survey," *Sensors*, vol. 23, no. 8, p. 4168, Apr. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/8/4168

[28] C. Zhao, J. Jiang, Y. Guan, X. Guo, and B. He, "EMR-based medical knowledge representation and inference via Markov random fields and distributed representation learning," *Artif. Intell. Med.*, vol. 87, pp. 49–59, May 2018, doi: 10.1016/j.artmed.2018.03.005.

[29] O.-J. Lee and J. J. Jung, "Story embedding: Learning distributed representations of stories based on character networks (extended abstract)," in *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, Sep. 2020, pp. 5070–5074, doi: 10.24963/ijcai.2020/709.

[30] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. 27th Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 26, Lake Tahoe, NV, USA, Dec. 2013, pp. 2787–2795. [Online]. Available: https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html

[31] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th Conf. Artif. Intell. (AAAI)*. Austin, TX, USA: AAAI Press, Jan. 2015, pp. 2181–2187. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571

[32] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell. (AAAI)*, vol. 28. Montreal, QC, Canada: AAAI Press, Jul. 2014, pp. 1112–1119. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531

[33] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*. Beijing, China: ACL, Jul. 2015, pp. 687–696, doi: 10.3115/v1/p15-1067.

[34] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proc. 30th Conf. Artif. Intell. (AAAI)*. Phoenix, AZ, USA: AAAI Press, Feb. 2016, pp. 985–991. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11982

[35] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proc. 32nd Conf. Artif. Intell. (AAAI)*. New Orleans, LA, USA: AAAI Press, Feb. 2018, pp. 1811–1818. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366

[36] Y. Tay, L. A. Tuan, M. C. Phan, and S. C. Hui, "Multi-task neural network for non-discrete attribute prediction in knowledge graphs," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, Nov. 2017, pp. 1029–1038, doi: 10.1145/3132847.3132937.

[37] Y. Wu and Z. Wang, "Knowledge graph embedding with numeric attributes of entities," in *Proc. 3rd Workshop Represent. Learn. NLP(RepNLP@ACL)*. Melbourne, VIC, Australia: ACL, Jul. 2018, pp. 132–136, doi: 10.18653/v1/w18-3017.

[38] Y. Li, X. Zhang, F. Wang, B. Zhang, and F. Huang, "Fusing visual and textual content for knowledge graph embedding via dual-track model," *Appl. Soft Comput.*, vol. 128, Oct. 2022, Art. no. 109524, doi: 10.1016/j.asoc.2022.109524.

[39] M. V. Yanaeva, E. V. Kuzminova, N. A. Akindinov, D. V. Osepchuk, and M. P. Semenenko, "Application of intelligent methods for diagnosing animal diseases based on determining the structures of blood facies," in *Proc. E3S Web Conf.*, vol. 262, 2021, p. 02005.

[40] T. Yamaguchi, K. Inoue, H. Tsunoda, T. Uematsu, N. Shinohara, and H. Mukai, "A deep learning-based automated diagnostic system for classifying mammographic lesions," *Medicine*, vol. 99, no. 27, 2020, Art. no. e20977.

[41] H.-J. Jeon, O.-J. Lee, and J. J. Jung, "Is performance of scholars correlated to their research collaboration patterns?" *Frontiers Big Data*, vol. 2, p. 39, Nov. 2019, doi: 10.3389/fdata.2019.00039.

[42] G. A. Gesese, R. Biswas, M. Alam, and H. Sack, "A survey on knowledge graph embeddings with literals: Which model links better literal-ly?" *Semantic Web*, vol. 12, no. 4, pp. 617–647, Jun. 2021, doi: 10.3233/SW-200404.

[43] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, p. 750, May 2020.

[44] Y. Shang, Y. Tian, M. Zhou, T. Zhou, K. Lyu, Z. Wang, R. Xin, T. Liang, S. Zhu, and J. Li, "EHR-oriented knowledge graph system: Toward efficient utilization of non-used information buried in routine clinical practice," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 7, pp. 2463–2475, Jul. 2021, doi: 10.1109/JBHI.2021.3085003.

[45] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017, doi: 10.1162/tacl_a_00051.

[46] X. Wang, X. He, Y. Cao, M. Liu, and T. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. Anchorage, AK, USA: ACM, Aug. 2019, pp. 950–958, doi: 10.1145/3292500.3330989.

[47] 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 (WWW)*. Florence, Italy: ACM, May 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.

[48] C. Jiang, M. Tang, S. Jin, W. Huang, and X. Liu, "KGNMDA: A knowledge graph neural network method for predicting microbe-disease associations," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 2, pp. 1147–1155, Mar. 2023, doi: 10.1109/TCBB.2022.3184362.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.

VAN THUY HOANG received the B.Sc. and M.Sc. degrees in computer science from Le Quy Don Technical University, Vietnam, in 2013 and 2018, respectively. He is currently pursuing the Ph.D. degree with The Catholic University of Korea, Republic of Korea. In 2014, he joined the Department of Telecommunication and Ciphers, Ministry of Public Security, Vietnam, as a Software Developer and a Teaching Assistant. He has published a number of papers at international conferences. His main research interests are graph representation learning, deep learning, machine learning, and information security.

THANH SANG NGUYEN received the bachelor's degree in software engineering from Duy Tan University, Vietnam, in 2021. He is currently pursuing the master's degree with The Catholic University of Korea, Republic of Korea. He was a DevOps engineer, from 2021 to 2022. He has published several papers at conferences. His research interests are deep learning, machine learning, graph representation learning, and AI applications.

SANGMYEONG LEE received the B.Sc. degree in physics from The Catholic University of Korea. He is currently pursuing the M.Sc. degree with the Department of Artificial Intelligence, Hanyang University, Republic of Korea. His main research interests are anomaly detection in time series and explainable artificial intelligence.

JOOHO LEE is currently pursuing the bachelor's degree with the School of Computer Science and Information Engineering, The Catholic University of Korea, Republic of Korea. His main research interests are graph representation learning, deep learning, and machine learning.

LUONG VUONG NGUYEN received the B.S. degree from the Department of Mathematics, Da Nang University of Education and Science, Vietnam, in July 2009, the M.S. degree from the Department of Computer Engineering, Da Nang University of Technology, Vietnam, in April 2013, and the Ph.D. degree in computer science and engineering from Chung-Ang University, Seoul, South Korea, in 2022. He has been a Lecturer and a Researcher with the Department of Artificial Intelligence, FPT University, Da Nang, Vietnam, since October 2022. He also has been an Adjunct Professor with the Department of Artificial Intelligence, The Catholic University of Korea, Bucheon, South Korea, since November 2022. His research topics include knowledge engineering in data science by using data mining, machine learning, ambient intelligence, and logical reasoning.

O-JOUN LEE (Member, IEEE) received the B.Eng. degree in software science from Dankook University, in 2015, and the Ph.D. degree in computer science and engineering from Chung-Ang University, in 2019. He has been an Assistant Professor with The Catholic University of Korea, Republic of Korea, since September 2021. Also, he was a full-time Researcher with the Pohang University of Science and Technology, Republic of Korea, from September 2019 to August 2021. His research topics are networked data analysis based on unsupervised/self-supervised representation learning and graph neural networks. He has applied the networked data analysis models and methods to various unstructured data, such as social media, bibliographic data, medical knowledge base, and traffic flow data.