IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 October 2022, accepted 30 November 2022, date of publication 20 March 2023, date of current version 13 April 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3227714

RESEARCH ARTICLE

A Knowledge Graph Embedding Framework With Triple Semantics

YAN WANG[1], HONGHAO LIANG[2], KAIHONG ZHENG[3], JINGFENG YANG[3], LUKUN ZENG[3], QIHANG GONG[3], SHENG LI[3], AND SHANGLI ZHOU[3]
[1]China Southern Power Grid Company Ltd., Guangzhou 510663, China
[2]Shenzhen Power Supply Bureau Company Ltd., Shenzhen 518001, China
[3]Digital Grid Research Institute, China Southern Power Grid, Guangzhou 510663, China

Corresponding author: Kaihong Zheng (zhengkh@csg.cn)

ABSTRACT The knowledge graph embedding model aims to use low-dimensional real-valued vectors to represent the entities and relations in the triples, where operations such as link prediction and triple classification can be performed based on these representations. However, existing embedding models only consider the structural embedding of triples, while ignoring the semantic information of triples. This paper proposes a knowledge graph embedding learning framework combined with triple semantic information (KGSE). KGSE comprehensively considers the structural embedding and semantic embedding of triples, where semantic embedding is used as a supplement to improve the quality of embedding. Specifically, KGSE uses the improved TransD model to obtain the structural embedding of triples, and employs the deep convolutional neural model combined with an attention mechanism to obtain the semantic embedding of triples. In addition, a novel energy function is designed to jointly train the above two embeddings. Experimental results show that the proposed framework improves significantly compared with Trans-based models and other baseline models in link prediction and triple classification tasks, which verifies the effectiveness of the proposed framework.

INDEX TERMS Knowledge graph, structure embedding, semantic embedding, link prediction, triple classification.

I. INTRODUCTION

Knowledge graph (KG) has attracted extensive attention because of its highly structured knowledge organization [1]. At present, representative knowledge graph bases (e.g., DBpedia, Freebase, and NELL) have become the core of many artificial intelligence applications, such as intelligent search, question answering, recommendation systems, and so on [2]. The knowledge stored in KG is in the form of triples, which makes machines difficult to well utilize this knowledge. The knowledge graph embedding model aims to use low-dimensional real-valued vectors to represent the entities and relations in the triples, where operations such as link prediction and triple classification can be performed based on these representations [3], [4], [5], [6], [7].

Currently, various knowledge graph embedding models have been proposed based on the symbolic representation of KGs with triples. Bordes et al. [8] proposed the TransE model, which embedded entities and relations by constraining the head entity vector plus the relation vector to be equal to the tail entity vector. Because of its simplicity and better performance, TransE has received widespread attention. However, it is precisely because of the simplicity that the predictive performance is not satisfactory for situations where the relation between the head entity and the tail entity is more complicated [9]. In order to deal with complex relations, Wang et al. [10] proposed the TransH model, which used two vectors (i.e., the translation vector and the normal vector) to represent the relationship. For each triplet in the knowledge graph, TransH projects the vector corresponding to the head entity and the tail entity to the hyperplane corresponding to the relationship along the normal vector, where the projection

The associate editor coordinating the review of this manuscript and approving it for publication was Ananya Sen Gupta.

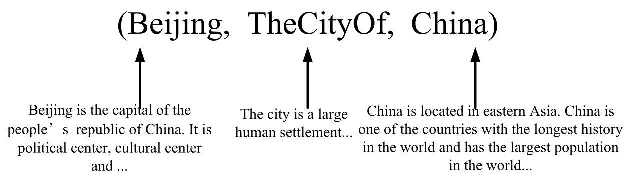

FIGURE 1. Example: semantic information description of triples in knowledge graph.

vectors meet the constraints of TransE. Since each entity has a different vector representation on different relationship hyperplanes, the representation of the entity is more flexible and differentiated. In this way, TransH can well learn the embedding representations of entities and relations. Lin et al. [11] proposed the TransR model, which held that entities and relations should not be in the same embedding space. TransR first maps the head entity and the tail entity to the space where the relation is located, and then establishes the translation constraints from the head entity to the tail entity. Due to the matrix multiplication operation, TransR not only has high complexity but also has a large number of parameters. Moreover, for any entity, TransR uses the same mapping matrix, which can not reflect the differences between entities and the characteristics of entities themselves. Ji et al. [12] proposed the TransD model, which represented the mapping matrix of TransR as three vectors. In TransD, there is a unique mapping matrix corresponding to each entity.

Although TransE and its variants can effectively learn embedding representations of entities and relations in the knowledge graph, they all simply learn the structural features of triples, while ignoring the rich semantic information of triples [13], [14]. For a sparse knowledge graph, it is easy to cause entities with different semantics to have similar embedding vector representations, which makes it difficult to break through its own performance bottlenecks [15], [16], [17]. For example, for such triples as $(Beijing, TheCityOf, China)$ and $(Shanghai, TheCityOf, China)$, when we use TransE to obtain the embedding vectors of entities and relations, we need to satisfy the two constraints of $T(Beijing) + T(TheCityOf) \approx T(China)$ and $T(Shanghai) + T(TheCityOf) \approx T(China)$. Finally, the vectors of $T(Beijing)$ and $T(Shanghai)$ will tend to be the same after training. To solve this problem, researchers tried to introduce more semantic features into the process of knowledge embedding [18], [19], [20]. Xie et al. [21] proposed the DKRL model, which used a continuous bag of words and a convolutional neural network to extract features from entity description. In addition, DKRL designed an energy function to integrate the entity description and the structural features extracted by TransE, thus improving the overall knowledge embedding performance. TKRL integrates the rich semantic information of entity types [22], while PtransE integrates the path information of the head entity to the tail entity [23]. Various approaches were proposed to improve the performance of knowledge embedding [24], [25], [26], [27], [28]. However, these approaches all use the TransE model to extract the structure embedding representation of triples, which will introduce the defect of TransE and ignore the semantic information of triples themselves. Actually, entities and relations in triples all have corresponding text descriptions, as shown in Fig. 1. Using more powerful and expressive model to improve the performance of structure embedding representation of triples is worth further exploring [29], [30], [31], [32]. Meanwhile, how to extract the semantic information of triples more effectively is challenging [33], [34], [35], [36], [37].

In order to represent and utilize knowledge more effectively, this paper comprehensively considers the extra semantic information of triples and embeds them into entities and relations, so as to learn a more complete knowledge embedding representation. Specifically, this paper proposes a knowledge graph embedding learning framework combined with triple semantic information (KGSE), which comprehensively considers the structural embedding and semantic embedding of triples. In the proposed KGSE framework, we use the improved TransD model to obtain the structural embedding of triples. We use the sentences containing entities and relations in triples as the auxiliary text so that we can effectively use the semantic information in the text to supplement the knowledge graph embedding learning. We employ the deep convolutional neural (DCN) model combined with an attention mechanism to obtain the semantic embedding of triples. Finally, a novel energy function is designed to jointly train the above two embeddings. Experimental results on datasets show that the proposed KGSE framework improves significantly compared with Trans-based models and other baseline models in link prediction and triple classification tasks, which verifies the effectiveness of the proposed framework. The contributions of this paper are summarized as follows:

- This paper proposes a knowledge graph embedding learning framework, which comprehensively considers the structural embedding and semantic embedding of triples. A mutually reinforcing energy function is designed to train the above two embeddings.
- This paper designs a more efficient and concise deep convolutional neural model based on attention mechanism to obtain the semantic embedding of triples, which is used as a supplement to improve the quality of embedding.
- Experimental results show that KGSE improves significantly compared with Trans-based models and other baseline models, which verifies the effectiveness of the proposed framework.

The remainder of this paper is organized as follows. We first review the related work in Section II and then describe a detailed description of our method in Section III. Section IV reports comprehensive experiments and comparison results. Finally, conclusions are presented in Section V.

II. RELATED WORK

This section summarizes the related work of knowledge graph embedding. We briefly discuss the proposed KGSE framework and existing methods that are most related to this work from different perspectives.

A. KNOWLEDGE EMBEDDED LEARNING

The initial knowledge graph is usually symbolic text description data, and we need to vectorize it so that the computer can use it for various tasks [38], [39], [40]. Knowledge embedded learning is trying to learn a projection from symbolic space to low-dimensional vectors, such as TransE and its variants [41], [42], [43], [44]. For each triple (h, r, t) in the knowledge graph, the vector u_r corresponding to the relation r is regarded as the translation from the vector u_h corresponding to the head entity to the vector u_t corresponding to the tail entity t, which satisfies the constraint $u_h + u_r \approx u_t$. TransE defines the triple score function as,

$$f_r = |u_h + u_r - u_t|_{L_1/L_2}, \tag{1}$$

where L_1/L_2 is the L_1 norm or L_2 norm of $|u_h + u_r - u_t|$. In the training process, TransE uses the maximum interval method to distinguish head entity vector u_h, relation vector u_r and tail entity vector u_t. Therefore, the optimization objective function is defined as,

$$L = \sum_{(h,r,t)\in S} \sum_{(h,r,t)\in S^-} max(0, f_r(h, r, t) + \gamma - f_{r'}(h', r', t')), \tag{2}$$

where γ is the interval between positive and negative triples, S and S^- are the sets of positive and negative triples, respectively.

The predictive performance of TransE is not satisfactory for situations where the relation between the head entity and the tail entity is more complicated. To address this problem, Wang et al. [10] proposed TransH model, which used two vectors (i.e., the translation vector l_r and the normal vector w_r) to represent a relation r. For each triple (h, r, t), TransH projects the vector u_h and u_t corresponding to the head entity h and the tail entity t to the hyperplane corresponding to the relation r along the normal vector w_r, where the projection vectors $u_{h_\perp}$, $u_{t_\perp}$ and l_r meet the constraints of TransE, i.e., $u_{h_\perp} + l_r \approx u_{t_\perp}$. Specially, $u_{h_\perp} = u_h - w_r^T u_h w_r$, $u_{t_\perp} = u_t - w_r^T u_t w_r$. Since each entity has a different vector representation on different relation hyperplanes, the representation of the entity is more flexible and differentiated, which enables TransH to better learn the embedding representations of entities and relations.

Lin et al. [11] proposed the TransR model, which held that entities and relations should not be in the same embedding space. For each triplet (h, r, t), TransR first maps the head entity h and the tail entity t to the space where the relation r is located, and then establishes the translation constraints from the head entity to the tail entity. Suppose h_r is the projection of head entity h in relation space, t_r is the projection of tail entity in relation space, and M_r is the mapping matrix from entity space to relation space. The specific mapping process of head entity and tail entity is as, $U_{h_r} = u_h M_r$, $U_{t_r} = u_t M_r$. Due to the matrix multiplication operation, TransR not only has high complexity but also has a large number of parameters. Moreover, TransR uses the same mapping matrix for any entity, which can not reflect the differences between entities.

Ji et al. [12] proposed the TransD model, which represented the mapping matrix of TransR as three vectors. In TransD, there is a unique mapping matrix corresponding each entity. Specially, for each triplet (h, r, t), the mapping matrix M_{rh} of the head entity h is defined as,

$$M_{rh} = r_p h_p + I^{m\times n}. \tag{3}$$

The mapping matrix M_{rt} of tail entity t is defined as,

$$M_{rt} = r_p t_p + I^{m\times n}, \tag{4}$$

where $r_p \in R^m$ is the projection vector of relation r, h_p and t_p are the projection vectors of the head entity h and the tail entity t, and I is the identity matrix.

Although the above models can effectively learn embedding representations of entities and relations in the knowledge graph, they all simply learn the structural features of triples, while ignoring the rich semantic information of triples [45], [46], [47]. In fact, the triples in the knowledge graph have a corresponding semantic introduction, which is very helpful for embedding learning. This paper proposes a knowledge graph embedding learning framework, which comprehensively considers the structural embedding and semantic embedding of triples.

B. MULTI-SOURCE INFORMATION INTRODUCTION

At present, most translation-based models focus on using the structural features of triples in knowledge graphs for representation learning, while ignoring the rich semantic information of triples. We can enhance the performance of knowledge embedding representation by introducing multi-source information. The introduction of multi-source information means that we should consider the structure information of the triple and other semantic information related to the triple (e.g., entity description information, entity type, and content information of the text where the entity is located) when learning knowledge embedding representation. The researches show that the introduction of multi-source information can significantly improve the performance of knowledge embedding [48], [49], [50].

Xie et al. [21] proposed the DKRL model, which integrated entity description based on TransE to improve knowledge embedding. Specifically, for entities in triples, DKRL uses structure-based embedding representation and entity description-based text representation to learn the embedding representation of entities in triples at the same time. The structure-based representation is the entity vector representation obtained by using the TransE model. This representation can effectively capture the structure information of the triplet. Entity description-based representation is the text representation vector obtained by using a continuous bag of words and a convolutional neural network, which can effectively capture the semantic information of the entity.

In order to integrate the two representations effectively, DKRL defines the energy function as follows,

$$E = E_s + E_d, \tag{5}$$

where $E_s = |h_s + r - t_s|$, h_s and t_s are the structure-based representation vector of head entity and tail entity, respectively. r is the representation vector of relation, and E_d is the energy function based on entity description, which is defined as follows,

$$E_d = E_{dd} + E_{ds} + E_{sd}, \tag{6}$$

where $E_{dd} = |h_d + r - t_d|$, $E_{ds} = |h_d + r - t_s|$, $E_{sd} = |h_s + r - t_d|$, h_d and t_d are the entity description-based representation vector of head entity and tail entity, respectively.

Although DKRL successfully integrates entity description information to improve the performance of knowledge embedding. DKRL uses the TransE model to obtain the structure representation of knowledge, which also introduces the shortcoming of the TransE model. Similar to DKRL, in the process of triple embedding learning, we also use structure embedding and semantic embedding of triple to learn the representation of entities and relations in the triple. Different from DKRL, we use structure-based embedding and triple context-based embedding. We use the improved TransD model to obtain the structural embedding of the triples, and use DCN combined attention mechanism to obtain the contextual semantic information of the triple. Meanwhile, it also should be noted that DKRL directly performs on the structure vector of the entity and the entity description vector, while our model first maps the structure vector and the semantic vector to the relation space before performing related operations. In addition, TKRL [22] model integrates rich semantic information of entity type, PtransE [23] integrates path information of head entity to tail entity. Various approaches are proposed to improve the performance of knowledge embedding. However, these approaches all use the TransE model to extract the structure embedding representation of triples, which will introduce the defect of TransE and ignore the semantic information of triples themselves. Bosselut et al. [51] introduced COMET model for automatic construction of commonsense knowledge bases. COMET is a framework for adapting the weights of language models to learn to produce novel and diverse commonsense knowledge tuples. Wang et al. [52] proposed a novel method of jointly embedding knowledge graphs and a text corpus so that entities and words/phrases are represented in the same vector space. Wang et al. [27] presented a text-enhanced knowledge embedding (TEKE) method for knowledge graph representation learning, which greatly expands the semantic structure of the knowledge graph. Rezayi et al. [53] proposed EDGE model to enrich knowledge graphs and node embeddings by exploiting auxiliary knowledge sources. In our model, instead of using entity description or words/phrases, we use semantic description of whole triplet as additional training text to improve the embedding quality. This paper proposes a convolution neural network model combined with attention mechanism to extract the semantic embedding of triples, and designs a new energy function to jointly train the model. Mai et al. [54] developed an entity retrieval system based on paragraph vectors and knowledge graph embeddings, which is applied in the bibliography field. Compared with them, we propose a network model with stronger ability to extract semantic information, and design a joint energy training function to better train the model.

Overall, this paper proposes a knowledge graph embedding representation model, which introduces the context semantic information of triples to improve the performance of knowledge embedding. In the process of triple embedding learning, we also use structure embedding and semantic embedding of triple to learn the representation of entities and relations in the triple. We use the improved TransD model to obtain the structural embedding of the triples, and propose a DCN model combined attention mechanism to obtain the contextual semantic information of the triple. Meanwhile, it also should be noted that other models directly perform on the structure vector of the entity and the entity description vector, while our model first maps the structure vector and the semantic vector to the relation space before performing related operations. Finally, we design a novel energy function to make the above vectors influence each other during the training process, and jointly improve the embedding quality of the triples [55].

III. THE PROPOSED MODEL

This paper proposes a knowledge graph embedding learning framework, which comprehensively considers the structural embedding and semantic embedding of triples. Considering the complexity and accuracy of the model, the improved TransD model is used to obtain the structure embedding of triples, and the entity description embedding model combined with attention and DCN is used to obtain the semantic vector of entities [56]. Finally, the above vectors interact with each other through the loss function in the training to improve the embedding quality of triples. The framework of the proposed approach is shown in Fig. 2.

In order to describe the work of this paper more clearly, several key definitions are given as follows.

Definition 1 (Knowledge Graph): A knowledge graph is represented as a set of fact triples, KG $= \{(e_h, r, e_t)\}$, where $e_h, e_t \in E$, $r \in R$. E and R are a set of entities and relations, respectively. T_O and T_U denote the set of observed triples and unobserved triples, respectively. Each triple $(e_h, r, e_t) \in T_O$ indicates that there exists a relation r from head entity e_h to tail entity e_t.

Definition 2 (Structure-Based Embedding): h_s and t_s are structure-based embeddings for head and tail entities. They can be learned by translation-based models like TransD.

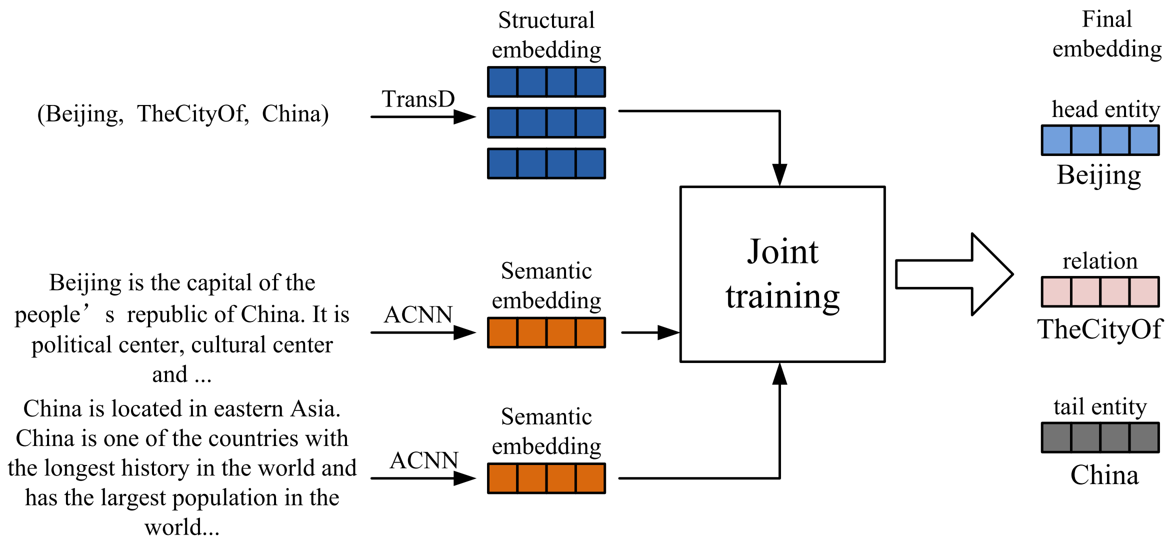

FIGURE 2. Framework of the proposed KGSE.

Definition 3 (Semantic-Based Embedding): h_d and t_d are the semantic-based embeddings for head and tail entities. They can be learned from semantic description of triples by following proposed ACNN model.

A. STRUCTURE-BASED KNOWLEDGE EMBEDDING

We declare that entity space refers to the space where entity vectors are located, relation space refers to the space composed of all relation vectors, and semantic description space refers to the space where semantic description vectors of triples are located. M_{rh} and M_{rt} are the mapping matrices that map the embedding vectors of the head entity h and the tail entity t to the relational space, respectively. M_{drh} and M_{drt} are the mapping matrices that map the description vectors of head entity h and tail entity t to the relational space, respectively. Inspired by TransD, the mapping matrix is calculated by three different vectors. The definitions of the above mapping matrices are as follows:

$$M_{rh} = r_p \times h_p + I, \tag{7}$$
$$M_{rt} = r_p \times t_p + I, \tag{8}$$
$$M_{drh} = r_p \times h_{des} + I, \tag{9}$$
$$M_{drt} = r_p \times t_{des} + I, \tag{10}$$

where r_p is the mapping vector related to relation r, h_p and t_p are the mapping vectors related to the head entity h and the tail entity t, respectively. h_{des} and t_{des} are the mapping vectors related to description text of the head entity h and the tail entity t, respectively. I is the identity matrix with diagonal element 1 and other elements 0.

In particular, the entity mapping process is shown in the following,

$$h_{e\perp} = M_{rh} \times h_s, \tag{11}$$
$$t_{e\perp} = M_{rt} \times t_s, \tag{12}$$
$$h_{d\perp} = M_{drh} \times h_d, \tag{13}$$
$$t_{d\perp} = M_{drt} \times t_d, \tag{14}$$

where h_s and t_s are the structure-based knowledge embedding vectors of the head entity h and the tail entity t, which are obtained by the improved TransD model. h_d and t_d are the semantic description-based knowledge embedding vectors of head entity and tail entity obtained by the proposed ACNN. The detailed process is shown in next sub-section.

B. SEMANTIC DESCRIPTION-BASED KNOWLEDGE EMBEDDING

Deep convolution networks (DCN) are known for their powerful local feature extraction capabilities, which are achieved by using convolution operations. Although the local feature extraction ability of DCN is very strong, it completely ignores the global features. This makes it difficult to capture the relevance between the long-term context information and discontinuous words, which further weakens the representation ability of sentences. In order to solve this problem and consider the actual application scenarios, this paper proposes an embedding model combining attention mechanism and DCN (named ACNN) to obtain semantic description-based knowledge embedding of triples [56], [57].

After word segmentation and word vector representation, the head entity description text can be transformed into a word vector sequence $(x_1, \cdots, x_n)$, where the k-th position is the entity vector x_k corresponding to the description text. The attention model takes the word vector sequence as the input, and the weight α_i is calculated as follows:

$$\alpha_i = \frac{exp(score(x_i, x_k))}{\sum_{j \neq k} exp(score(x_k, x_j))}, \tag{15}$$
$$score(x_i, x_k) = v_a^T tanh(W_a[x_i; x_k]), \tag{16}$$

where $score(x_i, x_k)$ is used to calculate the semantic relevance of entity vector and other word vectors, which is calculated by a simple two-layer BP neural network. In this network, W_a and v_a^T are the parameters of the first layer and the second layer, respectively. $tanh$ is used as the activation function. $[a; b]$ means to concatenate vector a and vector b.

After the weight score is calculated by attention, we can get the representation vector g of entity description text,

$$g = x_k + \sum_{i \neq k} \alpha_i x_i. \tag{17}$$

At the same time, entity description vector sequence $(x_1, \cdots, x_n)$ is also used as the input of DCN. In order to facilitate the calculation, we use a convolutional neural network architecture similar to DKRL. Through the feature extraction of DCN, the entity description vector f is obtained. Finally, the representation vector g obtained by attention model and the representation vector f obtained by DCN are concatenated and input into BP neural network to obtain the description embedding representation vector h_d of the head entity h,

$$h_d = W_b[f; g], \tag{18}$$

where W_b is the parameter matrix of BP neural network, $[f; g]$ means that f and g are concatenated.Similarly, we can obtain the semantic description-based embedding t_d of tail entity t and the semantic description-based embedding r_d of relation r.

After obtaining the structure-based embedding and semantic description-based embedding of triple, we design a novel energy function to jointly train the above two embeddings. Finally, knowledge embeddings of head entity, tail entity, and relation in the triple are learned, respectively. The detailed process is shown in the next sub-section.

C. ENERGY FUNCTION AND TRAINING PROCESS

In order to better integrate the structural-based and semantic description-based embedding of triples, we design the following energy function,

$$E = E_{de} + E_{dd} + E_{ed} + E_{ee} + E_{rr}, \tag{19}$$

where $E_{de} = ||h_{d\perp} + r_t - t_{e\perp}||$, $E_{de} = ||h_{d\perp} + r_t - t_{d\perp}||$, $E_{ed} = ||h_{e\perp} + r_t - t_{d\perp}||$, $E_{ee} = ||h_{e\perp} + r_t - t_{e\perp}||$, and $E_{rr} = ||r_t - r_d||$.

The model uses the negative triple sampling method for training. Specially, for each positive triple (h, r, t) in the training set, several negative triples (h', r, t) or (h, r, t') are constructed by replacing the head entity or the tail entity. In the process of training, the score of positive triples is as high as possible, and the score of negative triples is as low as possible. Therefore, the training objective function is defined as

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r',t')\in S^-} max(0, E(h, r, t) + \gamma - E(h', r', t')), \tag{20}$$

where S is the set of all positive triples, S^- is the set of all negative, $E(h, r, t)$ is the scoring function that is used to measure the distance between $h + r$ and t, and γ is a hyper-parameter. For the training of objective function, how to construct negative example triples is a key problem. In the existing research, there are two ways to sample negative triples, uniform negative sampling and preferred negative sampling. Uniform negative sampling refers to replace the head and tail entities with the same probability. Because of the complex relations in the knowledge graph, such as one-to-many, many-to-many, and many-to-one relations, this sampling method may lead to negative triples that are still correct. In order to improve this situation, a preferred negative sampling method is proposed. Specifically, for each relation r, t_{ph} is used to represent the number of tail entities in the knowledge graph after head entity h is fixed, and h_{pt} is used to represent the number of head entities in the knowledge graph after tail entity t is fixed. In preferred negative sampling method, the head entity is replaced by the probability of $t_{ph}/(t_{ph} + h_{pt})$, and tail entity is replaced by the probability of $h_{pt}/(t_{ph} + h_{pt})$. In short, if there are many kinds of tail entities in all triples containing r, the head entity is replaced by probability and vice versa. In this paper, the preferred negative sampling method is used to construct the negative triples. The pseudo-code of KGAE in the training phase is shown in Algorithm 1.

Algorithm 1 Knowledge Graph Embedding Learning Model Based on Structure and Auxiliary Information

Input: Knowledge Graph $G = \{(h, r, t)\}$, text auxiliary information $D = s$, entity set $\mathcal{E}$, relation set $\mathcal{R}$, embedding dimension k.
Initialization.
$e \leftarrow uniform(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$ for each entity $e \in \mathcal{E}$.
$r \leftarrow uniform(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$ for each entity $r \in \mathcal{R}$.
$w \leftarrow word2vec(w)$ for each entity $r \in \mathcal{R}$.
Training.
$T_{batch} \leftarrow \emptyset$
for each batch $S_{batch} \in G$ **do**
  for each triple $(h, r, t) \in S_{batch}$ **do**
    $H^- \leftarrow$ sample $(\{h'|(h', r, t) \notin G\})$
    $R^- \leftarrow$ sample $(\{r'|(h, r', t) \notin G\})$
    $T^- \leftarrow$ sample $(\{t'|(h, r, t') \notin G\})$
    $D \leftarrow \{s|m_h, m_t \in s\}$
    $T_{batch} \leftarrow T_{batch} \cup ((h, r, t), (H^-, R^-, T^-), D)$
  end for
  Update embedding by the gradient of Eq. (20).
end for
Output: The embedding representations of entities and relations in triples.

IV. EXPERIMENTS

In this section, we describe the experiments in detail to verify the effectiveness of our method. We first describe the data

TABLE 1. Statistics of different datasets.

Data Set	$\|Entities\|$	$\|Relations\|$	$\|Triplets\|$
FB15k	14951	1345	592213
FB15K-237	14541	237	310116

sets and experimental setup. Then the performance of the model is evaluated from two tasks, e.g., link prediction and triple classification. Finally, we perform ablation experiments and sensitivity analysis to further test the performance of the approach.

A. DATASETS AND EXPERIMENTAL SETUP

The knowledge graph data set used in this paper comes from Freebase. Freebase is a widely used knowledge graph, which contains dense entities and relations. At the same time, the triples have rich context information, which is suitable for our model. We conduct experiments on the following knowledge graph datasets: (1) FB15k; (2) FB15k-237. FB15k is a subset extracted from Freebase, in which all entities appear in the Wikilinks database. FB15k contains 592213 triples, including 14951 entities and 1345 relations. FB15k-237 is a subset of FB15k, which consists of relations from different domains like sports, people, locations and films, etc. We remove 47 entities without proper entity description from FB15k, and also remove all triples related to these entities. It contains 310116 triples, including 14541 entities and 237 relations. Some detailed statistics are described in Table 1.

This paper uses NYT100 as a text corpus database, which annotates entities and relations in Freebase. According to the entity pairs in FB15k and FB15k-237, the sentences containing these entity pairs are extracted from NYT100 as training text. For FB15k, 194385 sentences are extracted from NYT100, and these sentences are labeled according to the relationship between these entities in the knowledge graph. The annotated sentence covers 47103 triples, including 6053 entities and 699 relationships. For FB15k-237, 78978 sentences are extracted from NYT100, and these sentences are labeled according to the relations between these entities in the knowledge graph. The annotated sentence covers 6204 triples, including 3000 entities and 70 relations. When the description was counted, it is found that each entity or relation description in triples contains 69 words on average, and the longest description contained 343 words.

1) BASELINES AND IMPLEMENTATION DETAILS

We compare the proposed algorithm with the following methods: TransE[1] [8], TransH[1] [10], TransR[1] [11], TransD[1] [12], DKRL [21], TKRL [22], ConvE,[2] GAKE [38], CTransR,[3] PTransE [43], pTransE [52], TEKE [27], EDGE [53], TransP [58], AMCNN [59], CRAN [60], and GAATs [61]. For TransE, TransH, TransD and TransR, we learn a separate embedding matrix using the positive training entity pairs. Specifically, we select their learning rates for stochastic gradient descent in {0.001, 0.01, 0.05}, the margin in {0.5, 1, 2, 4}, the dimensions of entity and relation embedding in {20, 50, 100, 150}. We train TKRL model with mini-batch stochastic gradient descent. We select the batch size in {240,1200}, and margin in {0.5,1.0,1.5,2.0}. The dimensions of entity and relation are set to 100. All these embeddings are trained for 2000 epoches. For PTransE, we select the margin in {1,2,4}, the dimension of vectors in {50,80,100}, and the learning rate for stochastic gradient descent in {0.1,0.01,0.001}. For other methods, we used the implementation released by the corresponding authors with their best-reported hyperparameter settings or the results presented in their papers. The link prediction task is to predict the missing head entity or tail entity in a given incomplete triple. In link prediction, all entities are generally regarded as potential candidates, and the entity with the smallest score function is selected as the prediction entity. Because when the triple is incomplete, there may be more than one entity that predicts correctly. For example, when the tail entity is missing and r is a one-to-many relation, there may be several entities that are correctly predicted. Then, when evaluating a single triple, we need to remove the other correct answers and then recalculate the ranking. We use "fliter" to represent this practice and "raw" to represent the original practice. Meanwhile, we use "unif" to represent uniform negative sampling and "bern" to represent preferred negative sampling.

For evaluation, in the link prediction task, we use mean reciprocal rank (MRR) and Hits@k (the proportion of ranks no larger than k, k = 1, 3, 10) as criteria to compare different algorithms, which are standard metrics for knowledge graph completion tasks. In the triple classification task, we use accuracy (ACC) as the evaluation criteria.

B. LINK PREDICTION TASK

We first test the performance of the proposed algorithm on the link prediction task. In the experiment, the structure and semantic embedding dimensions of entities and relations are selected in {50,100,150,200}, the number of samples of negative triples is selected in {1, 2, 4, 6, 8}, the learning rate L_r is selected in {0.1,0.01, 0.001}, the global spacing value γ is selected in {0.5, 1, 1.5, 2}, and the batch-size B is selected in {200,500,1000}. The optimal hyper-parameters are determined by the experimental results on the validated set. After several experiments, the optimal parameters of this model are as follows: the structural embedding dimension of entity and relation is 100, the semantic embedding dimension of entity is 150, the number of negative triples is 6, the batch size is 500, the global gap γ is 1, and the learning rate L_r is 0.001. For the comparison algorithm, we select the parameters in the corresponding paper and the published code and select the experimental results after 1000 iterations for comparison. The experimental results are shown in Table 2.

[1]https://github.com/thunlp/Fast-TransX
[2]https://github.com/TimDettmers/ConvE
[3]https://github.com/Mrlyk423/Relation_Extraction

TABLE 2. Experimental results of link prediction.

Methods \ Datasets		FB15k								FB15k-237							
		MRR		hits@1		hits@3		hits@10		MRR		hits@1		hits@3		hits@10	
		raw	filter	raw	filter	raw	filter	raw	filter	raw	filter	raw	filter	raw	filter	raw	filter
TransE	unif	0.0040	0.0057	5.23	7.36	10.2	13.3	32.4	35.3	0.0121	1.1033	16.4	20.8	22.3	26.5	57.2	65.6
	bern	0.0043	0.0059	6.27	9.12	12.3	14.6	30.3	39.8	0.0136	0.0144	15.6	17.7	20.6	24.1	55.1	64.3
TransH	unif	0.0047	0.0054	6.54	9.31	12.5	15.3	33.7	37.3	0.0133	0.0144	16.1	18.2	19.5	22.6	53.1	62.6
	bern	0.0050	0.0058	7.63	10.6	13.1	16.2	37.2	38.2	0.0141	0.0148	16.9	17.9	20.4	21.1	54.2	63.7
TransR	unif	0.0044	0.0056	6.38	9.42	12.7	14.7	33.7	37.8	0.0134	0.0145	15.9	16.4	20.4	22.5	53.8	61.8
	bern	0.0050	0.0056	7.83	10.8	14.3	16.8	34.3	37.5	0.0143	0.0149	17.1	17.9	21.6	23.3	58.2	69.4
TransD	unif	0.0047	0.0060	6.58	9.73	12.7	14.3	32.5	34.3	0.0133	0.0142	16.8	17.4	20.1	22.4	54.8	64.3
	bern	0.0052	0.0061	7.93	9.63	13.5	15.7	33.1	35.7	0.0142	0.0152	17.7	18.9	21.3	23.2	53.7	60.9
ConvE	unif	0.0043	0.0052	5.38	7.52	10.2	15.6	37.3	44.5	0.0121	0.0136	15.2	18.4	22.4	26.1	42.4	54.1
	bern	0.0051	0.0058	6.32	8.67	13.3	17.2	36.7	39.8	0.0138	0.0151	16.8	19.5	22.6	25.5	50.3	61.5
pTransE	unif	0.0049	0.0051	8.26	11.2	13.7	12.1	21.4	33.2	0.0123	0.0137	16.4	20.8	23.2	25.2	39.5	53.3
	bern	0.0052	0.0058	7.54	9.31	13.7	17.4	38.2	43.6	0.0142	0.0153	16.3	23.1	24.3	26.6	46.8	59.2
DKRL	unif	0.0055	0.0059	6.42	8.52	13.2	15.2	38.6	40.2	0.0143	0.0149	16.8	19.0	21.5	22.8	58.3	63.1
	bern	0.0054	0.0058	7.67	9.72	15.3	15.9	39.3	42.8	0.0144	0.0150	18.5	20.4	20.7	23.1	62.3	68.4
TKRL	unif	0.0053	0.0058	10.4	12.2	14.7	16.3	40.1	43.2	0.0143	0.0147	20.4	22.6	23.5	26.2	60.5	68.3
	bern	0.0056	0.0061	12.5	14.5	15.7	18.4	41.2	45.2	0.0148	0.0151	22.3	24.1	26.6	27.9	61.5	69.2
KGSE	unif	**0.0061**	**0.0072**	**16.3**	**17.2**	**17.8**	**19.9**	**45.4**	**54.3**	**0.0154**	**0.0167**	**26.3**	**28.6**	**28.7**	**33.1**	**65.7**	**72.4**
	bern	**0.0068**	**0.0079**	**18.6**	**19.5**	**19.8**	**22.4**	**49.2**	**58.6**	**0.0162**	**0.0176**	**27.2**	**30.7**	**33.2**	**35.8**	**69.1**	**75.8**

[1] Bold indicates the best results.

It can be seen from Table 2 that the proposed method achieves the best experimental results, indicating that adding additional triple semantic description information can indeed improve the embedding performance of the knowledge graph. We use DCN combined attention mechanism to obtain the contextual semantic information of the triple. Meanwhile, it also should be noted that different from other methods, our model first maps the structure vector and the structure vector to the relation space before performing related operations. Finally, a mutually reinforcing energy function is designed to train the above two embeddings. From Table 2, we can also see that when using "bern" preferred negative sampling, the effect is improved significantly. At the same time, we can also see that for the same model, "bern" sampling is significantly better than using "unif" sampling under the MRR and hits@k evaluation criteria. This shows that "bern" sampling can construct more reasonable negative triples for training, which can improve the embedding effect of the model. In addition, we can also see that the performance of the proposed method in the "filter" case is better than that of "raw". This is because further filtering of the filter enhances the effect, indicating that there are many complex relations in the knowledge graph base, that is, one-to-many, many-to-one relations.

C. LINK PREDICTION TASKS UNDER DIFFERENT RELATION TYPES

According to the type of relations, it is divided into four types, 1-to-1, 1-to-many, many-to-1, many-to-many. For 1-to-1 relations, the head entity can only correspond to a tail entity, such as "spouse" relation. For 1-to-many relations, the head entity can correspond to multiple tail entities, such as "fatherOf" relation. For many-to-1 relations, a tail entity can correspond to multiple head entities, such as "nationality" relation. For many-to-many relations, multiple head entities can correspond to multiple tail entities, such as "friendOf" relation. For FB15k, the proportions of 1-to-1, 1-to-many, many-to-1 and many-to-many are 26.2%, 22.7%, 28.3%, 22.8%. We show the Hits@10 values in predicting head entity and tail entity under different relation types in Table 3.

It can be seen from Table 3 that the model in this paper surpasses most other models in predicting the head entity under many-to-1 and many-to-many relation types, and the tail entity under 1-to-many relation types. In other scenarios, the model in this paper can also be close to the best results. These results show that the model in this paper has advantages in complex relation prediction. The learned (entity and relation) embedding contains more information and is more discriminative.

D. TRIPLE CLASSIFICATION TASK

The triple classification task is to judge whether the given triple (h, r, t) is correct, which is essentially a binary classification task. In this experiment, the parameter selection is similar to the previous link prediction task. Specifically, the parameters of the model are set as follows: the structural embedding dimension of entity and relation is 100, the semantic embedding dimension of entity is 150, the number of negative triples is 6, the batch size is 500, the global gap γ is 1, and the learning rate L_r is 0.001. But the difference is that the number of negative samples is 1, and the semantic embedding dimension of entities and relation is 100. The experimental results are shown in Table 4.

Experimental results show that the accuracy of the approach proposed in this paper is better than other comparison algorithms on the triple classification task. Specifically, the proposed method achieves the optimal accuracy of 89.4% and 90.5% when using "unif" sampling method.

TABLE 3. Entity prediction under different relation mapping types.

Tasks	Prediction head entity (hits@10)				Prediction tail entity (hits@10)			
Methods	1-to-1	1-to-many	many-to-1	many-to-many	1-to-1	1-to-many	many-to-1	many-to-many
TransE (unif)	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransE (bern)	45.2	66.3	19.2	48.9	44.1	21.4	67.8	53.2
TransH (unif)	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH (bern)	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR (unif)	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR (bern)	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
CTransR (unif)	78.6	77.8	36.4	68.0	77.4	37.8	78.0	70.3
CTransR (bern)	81.5	89.0	34.7	71.2	80.8	38.6	90.1	73.8
TransD (unif)	80.7	85.8	47.1	75.6	80.0	54.5	80.7	77.9
TransD (bern)	86.1	95.5	39.8	78.5	85.4	50.6	94.4	81.2
TransP (unif)	81.5	86.7	52.3	85.3	81.1	61.3	84.2	84.6
TransP (bern)	84.6	95.2	47.2	87.6	83.9	59.4	94.5	86.6
PTransE (ADD, 2)	91.0	92.8	60.9	83.8	91.2	74.0	88.9	86.4
PTransE (MUL, 2)	89.0	86.8	57.6	79.8	87.8	71.4	72.2	80.4
PTransE (RNN, 2)	88.9	84.0	56.3	84.5	88.8	68.4	81.5	86.7
TEKE_H (unif)	66.6	80.9	58.0	79.6	60.5	60.4	81.5	80.2
TEKE_H (bern)	69.3	90.8	54.1	82.0	60.7	61.5	88.3	82.1
TEKE_R (unif)	66.2	82.0	57.0	81.3	62.5	57.5	83.1	81.2
TEKE_R (bern)	70.1	89.3	54.0	81.7	69.6	59.2	89.2	83.5
GAKE	90.7	90.3	**62.6**	**89.8**	87.3	**90.5**	89.9	83.1
Ours	**92.1**	**95.8**	61.2	85.5	**92.1**	80.3	**95.2**	**88.7**

[1] Bold indicates the best results.

When using "bern" sampling method, the proposed method achieves the optimal accuracy of 92.1% and 93.2%. It can be concluded that the proposed method can effectively integrate the semantic information of entity description into the embedding representation vector of entity to improve the discriminability of the entity. Hence, in the triple classification task, the proposed method can accurately judge whether the given triple is correct or not. In addition, it can be seen from the table that "bern" sampling can improve the accuracy of different models in the triple classification task, which shows that the negative triples obtained by sampling make the model training more reasonable and effective.

E. ABLATION STUDY

Further, we explore whether the entity and relation vectors extracted from different embedding models have an impact on our model. The experimental results using TransE, TransR, TransD, and TransH are shown in Table 5. Experimental results of Table 5 show that the entity and relation embedding vectors obtained by TransD are more conducive to the model in this paper.

We also explore the impact of the combination of different semantic description embedding methods on the performance of the model. Here, we first compare the general deep convolution network (DCN) and attention model, and then compare different attention-based convolution neural network models, such as AMCNN and CRAN. AMCNN is proposed by Liu et al. [59] for text classification. CRAN is introduced by Du et al. [60], which combines recurrent neural network and CNN-based attention model. The experimental results are shown in Table 6. The experimental results show that the deep convolution network combined with attention mechanism can better extract the description information of triples and integrate it into semantic embedding, so as to train with triplet structure embedding to learn better embedding representation. Specifically, the deep convolution network extracts the local features of semantic description, and the attention mechanism can integrate the global features to jointly improve the ability of information extraction. Although AMCNN and CRAN also have strong information extraction capabilities, the text semantic description of triples in the knowledge graph is relatively simple, so they are prone

TABLE 4. Experimental results of triple classification.

Datasets / Methods	FB15k		FB15k-237	
	unif	bern	unif	bern
TransE	77.3	79.8	79.3	80.2
TransH	74.2	79.9	76.3	77.1
TransR	81.1	82.1	80.2	83.5
TransD	86.4	88.0	88.3	89.4
DKRL	87.3	88.1	87.2	88.4
TKRL	86.5	88.4	88.6	90.6
GAATs	-	-	-	89.9
EDGE	88.6	89.7	90.1	91.4
KGSE	**89.4**	**92.1**	**90.5**	**93.2**

[1] Bold indicates the best results.

TABLE 5. Influence of different knowledge embedding methods on the model.

Methods	FB15k		FB15k-237	
	unif	bern	unif	bern
TransE	32.1	33.6	40.4	45.9
TransH	33.1	36.4	43.2	48.8
TransR	35.5	42.1	53.2	63.1
TransD	**49.2**	**58.6**	**69.1**	**75.8**

[1] Bold indicates the best results.

TABLE 6. Influence of different semantic description embedding methods on the model.

Methods	FB15k		FB15k-237	
	unif	bern	unif	bern
TransD + DCN	42.6	50.8	54.2	61.3
TransD + Attention	35.8	44.1	45.5	52.3
TransD + AMCNN	48.6	54.7	58.4	65.8
TransD + CRAN	47.6	55.4	57.4	66.2
TransD + ACNN	**50.2**	**58.3**	**60.1**	**69.7**

[1] Bold indicates the best results.

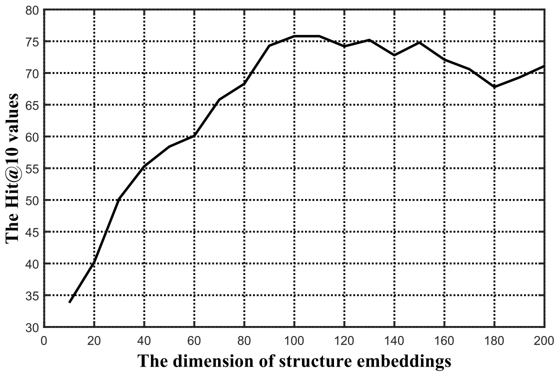

FIGURE 3. Sensitivity analysis of the structure embedding dimensions.

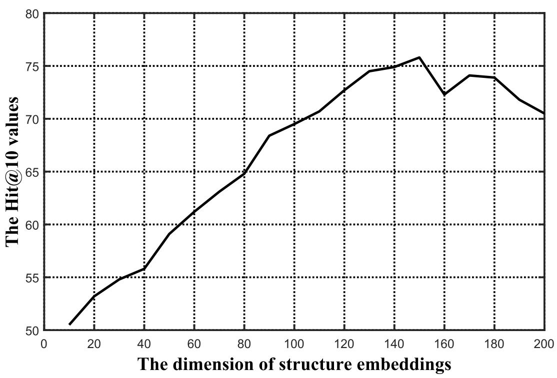

FIGURE 4. Sensitivity analysis of the semantic structure embedding dimensions.

to overfitting. At the same time, it is relatively difficult to adjust their parameters.

F. SENSITIVITY ANALYSIS

The proposed algorithm contains some important hyper-parameters, such as the dimension of embeddings, the number of negative triples and the global spacing value γ. We conducted experiments to explore the impact of these parameters on the performance of the proposed model. In the experiment, we found that the embedding dimensions of entities and relations in KG have a certain impact on the performance of the model. Hence, to illustrate the effect of embedding dimensions on the performance of the model, we conduct the hyper-parameter experiments on the FB15k-237 dataset. The results are shown in Fig. 3 and Fig. 4.

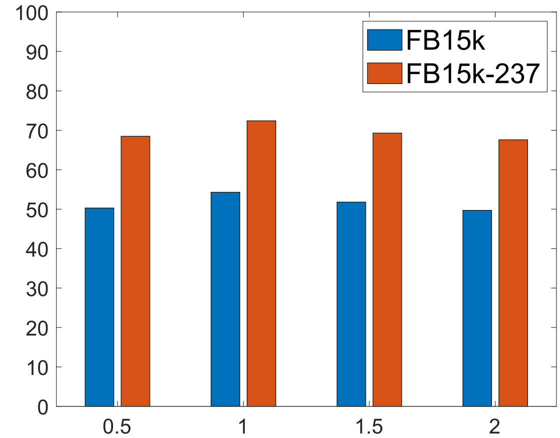

FIGURE 5. Sensitivity analysis of the global spacing value γ.

TABLE 7. Impact of the number of negative triples on the performance of the model.

	FB15k-237							
	MRR		hits@1		hits@3		hits@10	
	raw	filter	raw	filter	raw	filter	raw	filter
1	0.0078	0.0089	18.7	19.6	19.8	20.3	32.4	35.8
2	0.0103	0.0114	20.1	21.4	22.5	23.9	39.7	41.8
4	0.0127	0.0152	24.5	25.9	26.3	31.5	63.8	70.1
6	**0.0154**	**0.0167**	**26.3**	**28.6**	**28.7**	**33.1**	**65.7**	**72.4**
8	0.0138	0.0159	25.3	26.6	26.9	31.7	64.2	69.8

As can be seen from Fig. 3 and Fig. 4, when the dimension is low, the performance is generally low, because the embedding carries less information. With the increase of dimension, the performance of the proposed approach improves rapidly. This is because as the dimension increases, more information is obtained. When the dimension continues to rise, the results tend to stabilize in a certain range. Sometimes it even causes performance degradation because some redundant information is included in the learned embedding representation. For our experiments, the dimension of the entity and relation embedding vector is set to 100, and the dimension of the semantic description embedding vector is set to 150.

We conducted experiments on FB15k-237 dataset to test the impact of the number of negative triples on the performance of the model. The experimental results are shown in Table 7, where bold indicates the optimal model performance. From Table 7, we can see that when the number of negative samples is six, the model can get the best learning performance. Therefore, the number of negative triples was set to six in our experiments.

We conducted experiments on FB15K and FB15k-237 datasets to test the impact of the global spacing value γ on the performance of the model. In order to present the learning effect more conveniently, we use *hits*@10 as the performance evaluation criterion. The experimental results are shown in Fig. 5. From Fig. 5, we can see that when the global spacing value γ is set to 1, the model can get the best learning performance. Therefore, the global spacing value γ was set to 1 in our experiments.

V. CONCLUSION

The existing knowledge graph models ignore the semantic information of triples in the process of learning triples embedding. This paper proposed a knowledge graph embedding model based on auxiliary information, which comprehensively considers the semantic information of triples and integrates it into the representation of embedding vectors. The proposed model uses the improved TransD model to obtain the structural embedding of knowledge graph triples and uses the ACNN entity description embedding model to obtain the triple's semantic vector. By designing a novel energy function, the above vectors affect and promote each other in training. Finally, the comprehensive embedding representations of entities and relations are learned. Experimental results on knowledge data sets show that the proposed model significantly improves the embedding learning performance of triples. In the future, we will consider more auxiliary information, such as image information.

REFERENCES

[1] Q. Wang, Z. Mao, and B. Wang, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.
[2] Y. Zhang, H. Dai, Z. Kozareva, A. Smola, and L. Song, "Variational reasoning for question answering with knowledge graph," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 81–87.
[3] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022.
[4] R. Das, A. Neelakantan, D. Belanger, and A. McCallum, "Chains of reasoning over entities, relations, and text using recurrent neural networks," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, Long Papers*, 2017, pp. 132–141.
[5] W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A reinforcement learning method for knowledge graph reasoning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 564–573.
[6] G. Wan, S. Pan, C. Gong, C. Zhou, and G. Haffari, "Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning," in *Proc. Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1926–1932.
[7] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2168–2178.
[8] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.
[9] T. Trouillon, C. R. Dance, É. Gaussier, J. Welbl, S. Riedel, and G. Bouchard, "Knowledge graph completion via complex tensor factorization," *J. Mach. Learn. Res.*, vol. 18, no. 130, pp. 1–38, 2017.
[10] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1112–1119.
[11] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2181–2187.
[12] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 687–696.
[13] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, "Learning knowledge graph embedding with heterogeneous relation attention networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3961–3973, Aug. 2021.
[14] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proc. Int. Conf. Formal Ontol. Inf. Syst. (FOIS)*, 2001, pp. 2–9.
[15] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 90–96.
[16] S. He, K. Liu, G. Ji, and J. Zhao, "Learning to represent knowledge graphs with Gaussian embedding," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 623–632.
[17] B. Shi and T. Weninger, "ProjE: Embedding projection for knowledge graph completion," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1236–1242.
[18] J. Cheng, F. Zhang, and Z. Yang, "Knowledge graph representation learning with multi-scale capsule-based embedding model incorporating entity descriptions," *IEEE Access*, vol. 8, pp. 203028–203038, 2020.
[19] K. Ji, B. Hui, and G. Luo, "Graph attention networks with local structure awareness for knowledge graph completion," *IEEE Access*, vol. 8, pp. 224860–224870, 2020.
[20] J. Shi, H. Gao, G. Qi, and Z. Zhou, "Knowledge graph embedding with triple context," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 2299–2302.
[21] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2659–2665.
[22] R. Xie, Z. Liu, and M. Sun, "Representation learning of knowledge graphs with hierarchical types," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2965–2971.
[23] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 705–714.
[24] H. Gao, J. Shi, G. Qi, and M. Wang, "Triple context-based knowledge graph embedding," *IEEE Access*, vol. 6, pp. 58978–58989, 2018.
[25] M. Fan, Q. Zhou, T. F. Zheng, and R. Grishman, "Distributed representation learning for knowledge graphs with entity descriptions," *Pattern Recognit. Lett.*, vol. 93, pp. 31–37, Jul. 2017.
[26] J. Xu, X. Qiu, K. Chen, and X. Huang, "Knowledge graph representation with jointly structural and textual encoding," in *Proc. Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1318–1324.
[27] Z. Wang, J. Li, Z. Liu, and J. Tang, "Text-enhanced representation learning for knowledge graph," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 4–17.
[28] T. Bansal, D.-C. Juan, S. Ravi, and A. McCallum, "A2N: Attending to neighbors for knowledge graph inference," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4387–4392.
[29] C. Malaviya, C. Bhagavatula, A. Bosselut, and Y. Choi, "Commonsense knowledge base completion with structural and semantic context," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 2925–2933.
[30] S. Wang, S. Zhang, Y. Shen, X. Liu, J. Liu, J. Gao, and J. Jiang, "Unsupervised deep structured semantic models for commonsense reasoning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 882–891.
[31] X. Zhou, L. Niu, Q. Zhu, X. Zhu, P. Liu, J. Tan, and L. Guo, "Knowledge graph embedding by double limit scoring loss," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5825–5839, Dec. 2022.
[32] C. Ren, L. Zhang, L. Fang, T. Xu, Z. Wang, S. Yuan, and E. Chen, "Ontological concept structure aware knowledge transfer for inductive knowledge graph embedding," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
[33] H. Xiao, M. Huang, L. Meng, and X. Zhu, "SSP: Semantic space projection for knowledge graph embedding with text descriptions," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 3104–3110.
[34] Q. Zhu, X. Zhou, J. Tan, and L. Guo, "Knowledge base reasoning with convolutional-based recurrent neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2015–2028, May 2021.
[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[36] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[37] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 327–333.
[38] C. Li, X. Peng, Y. Niu, S. Zhang, H. Peng, C. Zhou, and J. Li, "Learning graph attention-aware knowledge graph embedding," *Neurocomputing*, vol. 46, pp. 516–529, Oct. 2021.
[39] V. Balogh, G. Berend, D. I. Diochnos, and G. Turán, "Understanding the semantic content of sparse word embeddings using a commonsense knowledge base," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 7399–7406.

[40] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. Eur. Semantic Web Conf.*, 2018, pp. 593–607.

[41] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2071–2080.

[42] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," 2019, *arXiv:1902.10197*.

[43] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 985–991.

[44] Y. Xue, Y. Yuan, Z. Xu, and A. Sabharwal, "Expanding holographic embeddings for knowledge completion," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4496–4506.

[45] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1955–1961.

[46] I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational Poincaré graph embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4463–4473.

[47] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 192–202.

[48] K. Toutanova, V. Lin, W.-T. Yih, H. Poon, and C. Quirk, "Compositional learning of embeddings for relation paths in knowledge base and text," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1434–1444.

[49] H. Xiao, Y. Chen, and X. Shi, "Knowledge graph embedding based on multi-view clustering framework," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 585–596, Feb. 2021.

[50] Y. Zuo, Q. Fang, S. Qian, X. Zhang, and C. Xu, "Representation learning of knowledge graphs with entity attributes and multimedia descriptions," in *Proc. IEEE 4th Int. Conf. Multimedia Big Data (BigMM)*, Sep. 2018, pp. 1–5.

[51] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, "COMET: Commonsense transformers for automatic knowledge graph construction," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4762–4779.

[52] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph and text jointly embedding," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1591–1601.

[53] S. Rezayi, H. Zhao, S. Kim, R. Rossi, N. Lipka, and S. Li, "Edge: Enriching knowledge graph embeddings with external text," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 2767–2776.

[54] G. Mai, K. Janowicz, and B. Yan, "Combining text embedding and knowledge graph embedding techniques for academic search engines," in *Proc. Semdeep/NLIWoD@ ISWC*, 2018, pp. 77–88.

[55] 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.

[56] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4710–4723.

[57] M. Zhao, W. Jia, and Y. Huang, "Attention-based aggregation graph networks for knowledge graph information transfer," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, 2020, pp. 542–554.

[58] F. Ren, J. Li, H. Zhang, and X. Yang, "TransP: A new knowledge graph embedding model by translating on positions," in *Proc. IEEE Int. Conf. Knowl. Graph (ICKG)*, Aug. 2020, pp. 344–351.

[59] Z. Liu, H. Huang, C. Lu, and S. Lyu, "Multichannel CNN with attention for text classification," 2020, *arXiv:2006.16174*.

[60] J. Du, L. Gui, R. Xu, and Y. He, "A convolutional attention model for text classification," in *Proc. Nat. CCF Conf. Natural Lang. Process. Chin. Comput.*, 2017, pp. 183–195.

[61] R. Wang, B. Li, S. Hu, W. Du, and M. Zhang, "Knowledge graph embedding via graph attenuated attention networks," *IEEE Access*, vol. 8, pp. 5212–5224, 2020.

YAN WANG received the master's degree from North China Electric Power University in 2015. He is currently a Senior Engineer with Guangdong Power Grid Co., Ltd. His research interests include electric energy measurement, electrical energy measurement automation systems, and electricity technology.

HONGHAO LIANG received the master's degree in engineering from Shanghai Jiao Tong University, China, in 2008. He is currently a Senior Engineer with Shenzhen Power Supply Bureau, China Southern Power Grid. His research interests include electric energy measurement, electrical energy measurement automation systems, electricity information acquisition systems, and software engineering.

KAIHONG ZHENG received the master's degree from Zhejiang University, China, in 2016. He is currently an Engineer with the Digital Grid Research Institute, China Southern Power Grid. His research interests include electric energy measurement, electrical energy measurement automation systems, electricity information acquisition systems, and electricity technology.

JINGFENG YANG received the master's degree from the South China University of Technology, China, in 2009. He is currently a Senior Engineer with China Southern Power Grid. His research interests include electrical energy measurement automation systems, electricity information acquisition systems, electric energy measurement, and big data analysis technology.

LUKUN ZENG was born in 1992. He received the B.S. degree in communication engineering and the Ph.D. degree in electrical engineering from North China Electric Power University, Beijing, China, in 2013 and 2019, respectively. He is currently working with the Digital Grid Research Institute, China Southern Power Grid. His research interests include electrical energy measurement automation systems and data mining technology.

QIHANG GONG received the master's degree from Zhejiang University, China, in 2020. He is currently an Engineer with the Digital Grid Research Institute, China Southern Power Grid. His research interests include electric energy measurement, electrical energy measurement automation systems, electricity information acquisition systems, and electricity technology.

SHENG LI received the master's degree from Hunan University, China, in 2019. He is currently an Engineer with the Digital Grid Research Institute, China Southern Power Grid. His research interests include electric energy measurement, electrical energy measurement automation systems, electricity information acquisition systems, and electricity technology.

SHANGLI ZHOU received the bachelor's degree from Tsinghua University, China, in 1991. He is currently a Professor level Senior Engineer with the Digital Grid Research Institute, China Southern Power Grid. His research interests include electric energy measurement, electrical energy measurement automation systems, electricity information acquisition systems, and electricity technology.