applied sciences

MDPI

Article

KTAT: A Complex Embedding Model of Knowledge Graph Integrating Type Information and Attention Mechanism

Ying Liu [1,2], Peng Wang [1,3,*] and Di Yang [1]

[1] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
[2] School of Computer Science, Tonghua Normal University, Tonghua 134002, China
[3] Changchun University of Science and Technology Chongqing Research Institute, Chongqing 401120, China
* Correspondence: wangpeng@cust.edu.cn

Abstract: Knowledge graph embedding learning aims to represent the entities and relationships of real-world knowledge as low-dimensional dense vectors. Existing knowledge representation learning methods mostly aggregate only the internal information of triplets and graph structure information. Recent research has proved that multi-source information of entities is conducive to more accurate knowledge embedding tasks. In this paper, we propose a model based on an attention mechanism and integrating the type information of entities, named KTAT. This model is based on the graph attention mechanism, to distribute corresponding attention mechanisms according to different weights between nodes. We introduce a type-specific hyperplane, which enables entities to have different embedding representations according to their type in the current triplet. Simultaneously, we also used textual description information of entities to improve the performance of the model. We conducted link prediction experiments on the FB15k and FB15k-237 datasets. The experimental results show that our model outperforms previous advanced methods compared to some baseline models, and demonstrate that combining type information can effectively improve the performance of link prediction.

Keywords: artificial intelligence; deep learning; knowledge graph embedding; graph attention mechanism; link prediction

check for updates

Citation: Liu, Y.; Wang, P.; Yang, D. KTAT: A Complex Embedding Model of Knowledge Graph Integrating Type Information and Attention Mechanism. *Appl. Sci.* **2023**, *13*, 7924. https://doi.org/10.3390/app13137924

Academic Editor: Ki-Yong Oh

Received: 31 May 2023
Revised: 2 July 2023
Accepted: 4 July 2023
Published: 6 July 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Knowledge graphs organize data into a graph structure from a semantic perspective, to establishing a comprehensive knowledge base that powers intelligent search services. An increasing number of large-scale knowledge graphs have been published on the internet recently, such as Freebase, DBpedia, and YAGO, and the deep integration of knowledge mapping technology and various industries has become an important trend. As the focus of knowledge graphs has grown in specific fields, domain knowledge graphs have also flourished in finance, health care, education, and other fields. A knowledge graph is a knowledge carrier of graph structure, where nodes represent entities and edges represent the interrelationships between nodes; by this means, it stores real things, concepts, and their relationships in the objective world. The principal assignment of embedding representation learning is to map the semantic information contained in the real-world things and transform it into low-dimensional dense vectors that are suitable for computational operations. Some studies, as well as expressing the information contained in the knowledge graph itself, have incorporated multi-source information to enrich the embedded representation of entities and relationships, usually including various forms of external information such as entity attribute information, text description, image information, etc. Integrating multi-source heterogeneous information adequately is of great significance for achieving knowledge embedding. It can simultaneously enrich the embedding information of entities

and relationships, and improve the sparsity problem of data, thus enhancing the representation ability of rare nodes and enhancing the separability of knowledge representation.

The entities in the knowledge graph always have multiple types, and each entity has different types in different roles. For example, the entity "Apple" may belong to the different types such as "Fruit" and "Company", and then the role played by "Apple" in different scenarios may also exist differently. Consequently, the vector representations of the entity "Apple" in the triplet (apple, established in, 1976) and triplet (apple, be rich in, vitamins) should be different. Therefore, distinguishing the different type representations of entities contributes to making knowledge embedding more accurate.

Several recent studies have incorporated entity type information into embedded representations. Krompaß et al. [1] integrated prior knowledge in the form of type constraints, treating entity types as hard constraints. The TKRL [2] (Type-embodied Knowledge Representation Learning) model integrates the hierarchical type information of entities and embeds types hierarchically to enrich entity information. KR-EAR [3] distinguishes existing knowledge graph relationships into attributes and relationships; it can learn representations of entities, attributes, and relationships to complete the knowledge completion task by separating relationship triplets and attribute triplets for modeling. Zhang et al. [4] utilized triplet information and entity attributes simultaneously for knowledge representation learning, encoding attribute information through convolutional methods, and fusing triplet information and attribute information to entity representations. Lin et al. [5] proposed a joint attribute preserving vector model for cross-language entity alignment, which embeds the vectors obtained from the two knowledge representations into a unified vector space, and further refines them by using the attribute correlation in knowledge embedding.

Combining text description information for embed representation learning could enrich the information expression by entity vectors. Several studies utilized pretraining language models to learn the semantic information contained in the knowledge graph, such as Bert [6] (Bidirectional Encoder Representations from Transformer), a multi-layer and bidirectional transformer encoder pretraining language model, which uses the semantic information contained in the text to embed representation learning through unlabeled corpus training. DKRL [7] (Description-embodied Knowledge Representation Learning) uses text description information as external information, using continuous word bags and deep convolutional neural network models to embed the semantics of entity descriptions. Inspired by relevant research, this paper proposes a knowledge embedding method based on an attention mechanism and incorporating entity types. It uses the convolution method to embed text description information to obtain the initialization vector, then assigns different weight values to entities under different roles according to different entity types, and then assigns different attention weight values to neighbors through the attention mechanism to complete neighborhood information aggregation. Our model is divided into three layers: initial embedding based on text information, entity fusion of type information under the current role, and fusion of the attention mechanism for graph representation learning.

(1) We propose a knowledge representation model named KTAT, which integrates type information and the attention mechanism. This model simultaneously encodes entities and relationships, mapping them to type-specific weight matrices based on the different types by entities in different triplets, and obtaining their new entity embedding vectors that contain their own specific attribute information. Our model also utilizes text description information to initialize the entity vector, thereby improving the initial representation and enhancing the representation ability for rare entities.

(2) We integrate the attention mechanism and type information into knowledge representation simultaneously, enabling the attention mechanism to be applied to knowledge graphs of heterogeneous neighborhoods, and we strengthen the node representation methods on this basis and embed entities and relationships.

(3) We conducted link prediction experiments on the FB15K and FB15K-237 datasets, and conducted in-depth analysis of parameters on KTAT. Experiments on both datasets

showed that the KTAT model outperformed existing advanced knowledge embedding models and performed well in most evaluation indicators.

2. Related Work

The current knowledge graph embedding methods can be roughly divided into two types based on whether they fuse multi-source information. The knowledge graph embedding method without fusing multi-source information mainly relies on the inherent information in triplets and graph structure for knowledge representation learning.

Classic knowledge embedding models are based on a single triplet, such as TransE [8] and its variants. TransE regards the relation as the translation distance between entities in vector space, and proposes the assumption that the tail entity in a triple minus the head entity is approximate to the vector representation of the relation. This simple but intuitive method has obtained effective results in the experiments for single relationships, but there are still limitations for the representation of 1-N and N-N relationships. The TransH [9] proposes that an entity should have different representations under different relationships. The initialization vector representation of the entity is projected onto the unit vector to obtain a new entity representation, but it still maps the entity and relationship to the same semantic space, which to some extent limits the representation ability. TransD [10] projects the head entity and tail entity into different vector spaces, respectively, and constructs the projection matrix dynamically through the projection vector, which greatly saves the amounts of parameters and calculations. TransR [11] believes that entities in different relationships have different semantic spaces, so entities and relationships are embedded in two different spaces; that is, entity space and multiple relationship-specific entity spaces, and the entities are projected into a shared hyperplane through the relationship projection matrix in the TransR model. TransM [12] and TransF [13] optimized the model by changing the evaluation function of TransE, improving its ability to represent complex relationships.

The knowledge representation learning method based on graph structure information, such as DeepWalk [14], mainly draws on the idea of Word2vec training word vectors and applies this idea to the data of graph structure information. The Word2vec [15] algorithm utilizes skip-gram and CBOW methods to train word vectors based on the co-occurrence relationship of words in the corpus, while DeepWalk uses random walk for node sampling to generate a text sequence, and then uses the skip-gram model to convert the sequence into embedding. Compared with the random walk strategy of the graph in DeepWalk, Node2vec [16] weighs the results of graph embedding by adjusting the probability of random walk. Node2vec believes that the word sequence generated by simple random walk cannot explain the relationship between nodes accurately, it optimizes the process of random walk, and designs a search strategy to control the search through parameter settings. Node2vec adjusts the parameters to generate the relationship between the nodes in the sequence generated by random walk, which effectively balances the homogeneity and isomorphism of embedding.

GCN [17] (Graph Convolutional Network) ingeniously designs a method to extract features from graph data, introduces the concept of convolution into heterogeneous graph structures, and aggregates neighbor information according to the Laplacian matrix. RGCN [18] (Relational Graph Convolutional Network) proposes a method for multi-relationship aggregation in heterogeneous graphs, applying the method of GCN to model relationship data in multi-relationship graphs with relationship weights, especially for link prediction and entity classification tasks. GraphSAGE [19] proposes a general inductive framework that learns the topological structure of neighborhoods of each node and the distribution of node characteristics in the neighborhoods. It trains a multi-level aggregate function to generate node embedding according to the local neighborhoods' sampling and aggregation characteristics of the node. GAT [20] (Graph Attention Network) utilizes the attention mechanism method to calculate and assign different attention weight values to each node in the graph based on the characteristics of neighboring nodes. Each operation of GAT needs to traverse all nodes on the graph, which breaks away from the constraint of

the Laplacian matrix, and therefore solves the limitation that GCN cannot handle dynamic graphs. HRAN [21] (Heterogeneous Relation Attention Network) integrates the attention mechanism and proposes an end-to-end heterogeneous relational attention network framework. It utilizes the attention mechanism to learn the importance of each relation path, selectively hierarchically aggregates the information characteristics of neighbors, and fuses each type of specific semantic information through the relationship path. RGHAT [22] (Relational Graph Neural Network with Hierarchical Attention) uses a hierarchical attention approach that explicitly utilizes the local neighborhood information of each entity, introducing a relational attention layer and an entity attention layer, distinguishing the importance of different relations and the importance of different adjacent entities under the same relation in a hierarchical structure. The hierarchical attention mechanism provides a fine-grained learning process for the model, which increases its interpretability.

The key task of knowledge representation based on multi-source information is data level fusion, the essence of which is to expand and fuse rich additional information related to the knowledge base, such as entity type information, text description information, logic rules, graph structure information, and other information. Integrating multi-source information of entities can effectively explore semantic correlations between entities, and more accurate semantic representations can improve the distinguishability of triplets. Simultaneously, it can also enhance the ability of the model to solve zero-shot problems.

Many large-scale knowledge graphs contain concise descriptions of entities, which contain numerous pieces of implicit information about the entities, and fully utilizing the text statements describing entities can better complete knowledge embedding. DKRL fuses the description text information of entities in the knowledge representation learning. It considers two models of CBOW and CNN as the embedding method of text description information to obtain the text-based entity embedding representation [7], then fuses the text information into the entity embedding through TransE training. Munne et al. [23], by utilizing the abstract and attribute information of entities in the knowledge graph, and learning the representation of entities through the relationship triplets, attribute triplets, and abstracts, then completed further entity alignment tasks. Cheng et al. [24] proposed TADW (Text-Associated DeepWalk), which proved that the DeepWalk method is actually equivalent to the matrix decomposition method, extending the matrix decomposition model to support embedding of text feature information. The CANE [25] model utilizes the rich external attributes of nodes and the mutual attention mechanism to model complex network relations; it fuses the structural and textual information of nodes and ultimately produces dynamic contextual vectors, allowing entities to have different representations when interacting with different neighborhoods.

Several studies fused entity attributes and type information to obtain more accurate embedding representations, and demonstrated the effectiveness of this method through experiments. SSE [26] adds constraints to the objective function, so that entities belonging to the same type have similar positions in the vector space to ensure semantic smoothness. TransT [27] proposes a method of entity embedding that integrates structured information and describes entity types, capturing the prior distribution of entities and relationships that utilize the type-based semantic similarity of related entities and relationships. This model utilizes the type-based prior distribution, multiple embedded representations of each entity are generated in different contexts, and the posterior probability and relationship prediction of entities are estimated. TKRL [2] is a knowledge representation learning model that utilizes entity type information as external information; entities of different types have different representations. For the hierarchical structure of entity types, TKRL uses two encoding types to modeling hierarchical structures, ultimately confirming that entity types can fulfil an important role in knowledge representation learning.

Lin et al. recognized the difference between relations and attributes, proposed to separate relation triples and attribute triples, designed a model KR-EAR [3] that could simultaneously represent the relation between entities and represent attributes of entities, optimized the objective function in the form of conditional probability, and trained by

maximizing the joint probability of relation triples and attribute triples embedded in entities. The experiments show that this model outperforms others in entity prediction, relationship prediction, and attribute prediction tasks. AKRL [4] (Attribute-embodied Knowledge Representation Learning) put forward the idea that relational triples and attribute triples have different meanings and should have different training methods. It regards the attribute triples as the attribute information of entities, and utilizes deep convolutional neural networks to learn the attributes of entities.

3. Methods

3.1. Overall Framework

In this section, we explain the technical details of the KTAT model in detail, our work constructs a knowledge learning framework that integrates attention mechanisms and entity type information, embedding entities and relationships in the knowledge graph into low-dimensional dense vectors. We record the knowledge graph as $G = (E, R, T)$, where the set of all entities is represented as $E = \left\{e_1, e_2, \ldots, e_{|E|}\right\}$, the relationship set is represented as $R = \left\{r, r_2, \ldots, r_{|R|}\right\}$, and the triples set is represented as $T_r = \{(e_h, r, e_t)|e_h, e_t \in E, r \in R\}$, where (e_h, r, e_t) represents a triple in the knowledge graph, e_h represents head entity, e_t represents the tail entity, and r represents the binary relationship. $e_h, r, e_t \in \mathbb{R}^d$ represents the head and tail entities and relationships in the triplet, $E \in \mathbb{R}^{|E| \times d}$ and $R \in \mathbb{R}^{|R| \times d}$ represent entities and relationships matrix, respectively.

3.2. Initialization Embedding Based on Text Information

We take the word vector representation pretrained by Word2vec and Wikipedia corpus as the initial input, treat the text description information of an entity as a sentence, and embed the entity description information as the initialization vector representation of an entity through convolution operation. The convolutional operation can better obtain the internal text information hidden in the word order, more accurately represent the semantics of the entity description text. It makes the initial embedding of entities more accurate and easier for subsequent training of the model.

In order to encode entity description information into a fixed length embedding vector, we set up two convolutional layers for encoding. Firstly, extract each word vector from the text description statement and record the set of all word vectors as $M_w = \{w_0, w_1 \ldots w_i\}$, where the initialization vector dimension of each word is 300, and the convolution kernel is defined as $C_w \in \mathbb{R}^{\theta \times \theta}$, and corresponding pooling layers are set for each convolutional layer. The first layer of convolution selects maximum pooling to preserve local features of the text due to the need to remove residual noise within the statement, while the second layer of convolution adopts average pooling. We regard the vector $h^{(0)} = \{\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots, \overrightarrow{h}_{|E|}\}$ as the initial embedding of the entity, which is obtained from the text description information, where $\overrightarrow{h}_i \in \mathbb{R}^d$ represents the feature vector representation of the i-th node, |E| represents the number of nodes, and d represents the vector with the dimension of $\overrightarrow{h}_i$.

3.3. Entity Type Information Fusion Layer

Due to the heterogeneity of knowledge graphs, the same entity in different types might appear in different feature spaces, directly aggregating all neighbor features of each entity usually unable to accurately represent the implicit and accurate semantics of the entity. Therefore, we believe that the entity should have different representation when it plays different roles in different relations under mapping of each type. In order to obtain the embedding of entities under different types, we constrain the types of head and tail entities through the type of the relation. We introduce attribute-specific weight matrices to map the entity to the attribute-specific plane, so that the entity has different embedding representations that combine type information under different types.

We construct a types triplets set $T_s = \{(s_h, r, s_t)|s_h, s_t \in S, r \in R\}$ for each relation, and distinguish it from relation triplets (e_h, r, e_t), where $S \in \mathbb{R}^{|S| \times d}$ represents the set of entity types, the embedding dimension of entity types is the same as that of entities, $s_h \in \mathbb{R}^d$ represents the type of head entity to which under a specific relationship, r belongs to the current triplet, $s_t \in \mathbb{R}^d$ represents the type to which the tail entities belong to. Figure 1 shows the difference between the relational triplet and the type triplet.

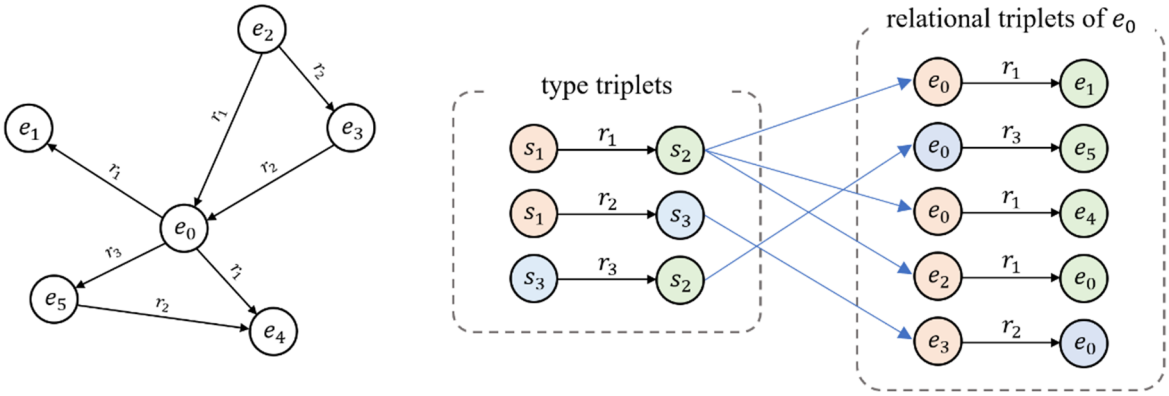

Figure 1. This figure is an example of relationship triplets and type triplets, where each color represents a kind of type information.

Before aggregating neighborhood node information, we first map the entity to a higher dimensional space, so as to obtain stronger expression ability.

$$z_i^{(l)} = \Phi_{sp}\left(h_i^{(l)}\right) = W_{sp}{}^{(l)}(h_i^{(l)} \cdot s_{h_i}{}^{(l)}) \tag{1}$$

where $\Phi_{sp}(\cdot)$ represents a mapping function of a type-specific plane, which obtains a new embedding representation of the entity in the current role in space while maintaining the embedding of the entity itself unchanged. We used a bitwise multiplication method to fuse entity vectors and entity type vectors in the experiment; using the feature matrix $W_{sp} \in \mathbb{R}^{d' \times d}$ performs a linear transformation on the node vector h to obtain a new node feature vector $z^{(l)}$, where $z_i^{(l)}$ represents the vector representation of the nodes processed in layer l, and d' represents the output dimension for entity $z_i^{(l)}$, and $z^{(l)} = \left\{\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_{|E|}\right\}$, $\vec{z}_i \in \mathbb{R}^{d'}$ is the output entity embedding representation that integrates type information.

3.4. Neighborhood Information Attention Aggregation Layer

Aggregating adjacent node information through message passing paradigms to update central node information is a key task of message passing graph neural networks, it extends convolutional operators to the field of irregular data to achieve information transfer between nodes. We used an attention mechanism messaging method to aggregate neighbor information, representing the correlation between node features by calculating the attention weight values between nodes and their neighbors.

We utilized the self-attention calculation method to calculate the weight values of the interaction between each pair of nodes, and utilized the concatenation method to map the vector $\vec{z}_i$ and $\vec{z}_j$, which mapped to attribute-specific space transformation into a $2d$-dimensional vector, where $\vec{z}_i$ represents the current node, $\vec{z}_j$ represents the neighbor of $\vec{z}_i$. We transformed the vector into a value that measures the weight of mutual influence between nodes based on the embedded features of the nodes; in this experiment, we selected the transposed vector $\vec{a}$ and the concatenated vector for inner product operation.

We obtained e_{ij} through the activation function, which is the attention weight of node j to node i.

$$e_{ij}^{(l)} = LeakyReLU(\vec{a}^{(l)^T}(z_i^{(l)}||z_j^{(l)})) \tag{2}$$

We selected LeakyReLU activation function in this experiment, an activation function of neural network unit, which can effectively solve the problems of vanishing gradient and death ReLU. This attention weight is the interaction weight between all node pairs, representing the importance between each node pair without considering the graph structure information. Knowledge graph is a directed graph; the weight value also varies due to the different positions of entities in the triplet, and e_{ij} is not equal to e_{ji}, this weight forms an asymmetric self attention matrix of $|E| \times |E|$ dimensions.

After obtaining the attention weight values between node pairs, they are normalized using softmax and then through masked attention to introduce attention weight into the graph structure.

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in N_{(i)}} \exp(e_{ik}^{(l)})} \tag{3}$$

where $\alpha_{ij}^{(l)}$ represents the normalized attention coefficients of the current node and its neighbor nodes. We aggregate the information of neighboring nodes and introduce a weight matrix W_{att}, obtaining the implicit representation of nodes through linear transform.

$$h_i^{(l+1)} = \Phi_{er}\left\{\sigma\left(\sum_{j \in N_{(i)}} \alpha_{ij}^{(l)} z_j^{(l)}\right)\right\} + h_i^{(l)} \tag{4}$$

We utilize the method of adding a vector representation of our own node that has been transformed by the weight matrix W to enhance the semantics of the node itself, $h_i^{(l+1)}$ contains information about the node itself and aggregated neighborhood information. In order to better aggregate neighborhood information, we aggregate the relation embedding vectors in neighbor paths into the current node and propose a relationship fusion function $\Phi_{er}(\cdot)$. The initial embedding representation of the relation is set to $r^{(0)} = \left\{\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_{|R|}\right\}$ through random initialization, and the relation is fused with the entity embedding according to the different positions of the current entity, as shown below:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N_{(in)}} \alpha_{ij}^{(l)} z_j^{(l)}\right) \star k\vec{r}_i + \sigma\left(\sum_{j \in N_{(out)}} \alpha_{ij}^{(l)} z_j^{(l)}\right) \circ k\vec{r}_i + h_i^{(l)} \tag{5}$$

We define $\star$ and $\circ$ as two operators with different operational functions, which consider the different directions of the relation in the graph. Based on the different positions of the current node at the beginning and end of the triplet, the connection paths of the current node are divided into two types. If the current node is at the end of the triplet and the relation with neighboring nodes points to the current node, the aggregation operator is set to $\star$; the strategy we adopted in the experiment was to add aggregated type information and features bitwise. If the current node is at the head node in a triplet and the relationship with the neighboring node points to the neighboring node, the aggregation operator is set to $\circ$, and in the experiment, it was set to subtract the correspond features bitwise. Where k is a hyperparameter to adjust the proportion of relational embedding vectors in the current path, we conducted further experiments and research on the value of K in order to achieve better embedding effect. Figure 2 shows the architecture of KTAT model.

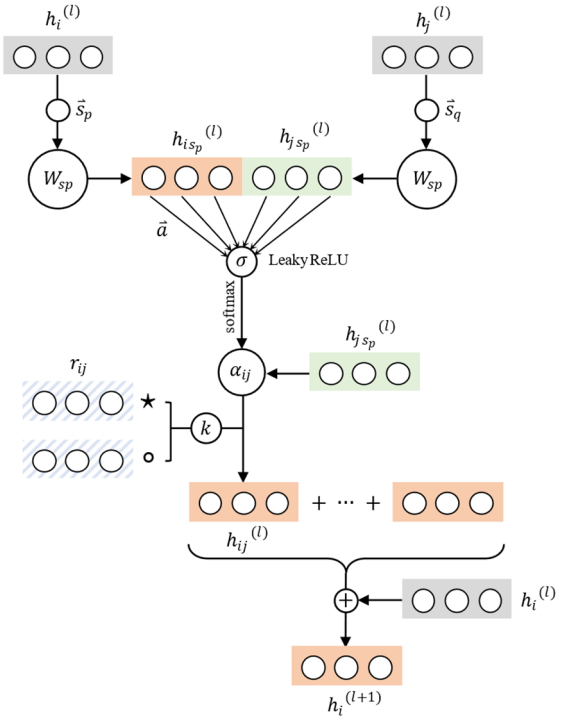

Figure 2. Schematic diagram of updating nodes and relations, where orange block represents the center node vector, green block represents the neighbor node vector, and blue block represents the relationship vector.

3.5. Training Objective

The connection prediction task of entities is the process of predicting adjacent entities specified by an entity with a specific relationship; we evaluated the performance of our model through link prediction experiments. For each triplet, the purpose is achieved by defining the general method of the scoring function $f(e_h, r, e_t)$. Based on the translation assumption, we set the scoring function as follows:

$$Loss = \sum_{(e_h,r,e_t)\in T} \sum_{(e_h\prime,r,e_t\prime)\in T\prime} [\gamma + d(e_h + r, e_t) - d(e_h\prime + r, e_t\prime)]_+ \quad (6)$$

The basic idea of setting the loss function is that the mapping of the head entity and the relation in the vector space should be as close as possible to the tail entity for the fact expressed by a correct triple; that means $d(e_h + r, e_t)$ in the correct triple should be as small as possible, conversely, $d(e_h + r,\ e_t)$ should be as large as possible for an incorrect triple, where the distance formula d is constrained by $L2$ regularization, and function d represents the distance between two vectors mapping to the vector space. Negative example triplet $(e_h' + r, e_t')$ is constructed by replacing one of the head or tail of the triplet with another random entity. The set of negative example triplet $T'_{(e_h,r,e_t)}$ is as follows:

$$T'_{(e_h\prime,r,e_t\prime)} = \{(e_h\prime, r, e_t)|e_h' \in E\} \cup \{(e_h, r, e_t\prime)|e_t' \in E\} \quad (7)$$

By minimizing the loss function *Loss*, we can minimize the distance of positive cases and maximize the distance of negative cases, so as to achieve the purpose of training the knowledge representation of entities and relations.

4. Experiment

4.1. Dataset

We conducted classification and prediction experiments on the FB15K [8] and FB15K-237 [28] datasets, respectively, and the statistical information of the dataset is shown

in Table 1. FB15k is a classic subset from the FreeBase knowledge graph learning field. Xie et al. [7]. collected all type instances and relationship-specific type information of entities in the FB15K dataset, and filtered out type instances that did not appear in the extracted relationship-specific type information. After filtering, each entity in the dataset had at least one type, resulting in a total of 571 entity types. We also used subset FB15k-237 in the experiment, which is a smaller but more refined subset based on the original FB15k dataset, and removed the opposite relationship.

Table 1. Statistics of FB15K and FB15K-237 datasets.

Dataset	Entities	Relations	Train	Valid	Test
FB15K	14,951	1345	483,142	50,000	59,071
FB15K-237	14,541	237	272,115	17,535	20,466

4.2. Experimental Setup

4.2.1. Evaluation Protocol

We used the embedded vector representation of entities and relationships trained by the model, and replaced the head or tail of the real triplet, respectively, to construct a set of broken triples. Then we calculated the distance between the head vector and tail vector in the correct triplet and the wrong triplet and mapped them to the vector space, ranking the distance from low to high. We used the proportion of correct entities with average rank (MR), average reciprocal rank (MRR), and top N rankings (Hits@1 Hits@3 Hits@10) as indicators, the data we reported were set as "filtered".

4.2.2. Implementation Details

We used the Adam optimizer for batch training to accelerate the training and convergence speed of the model, with the batch size among {64, 128, 256}, and the embedding dimension of the entity and relation among {50, 100, 150, 200}. We set two CNN convolutional layers to obtain the initial embedding of the text description information, with convolutional kernels set to 5×5 and 3×3, respectively, corresponding to maximum pooling and convolutional pooling; the initial vector of the relation was randomly initialized. The number of layers L was among {1, 2, 3}, which means that each iteration was aggregated L times node by node, with the learning rate among {0.01, 0.001, 0.005}. We used dropout to prevent overfitting in our model, and dropout was sampled from {0.1–0.5}. The value k of the relation weight was sampled from {0–1}, and we conducted further experiments and analysis with a step size of 0.1. We regarded MRR as the evaluation label and selected the best configurations of parameters as: batch size = 256, embedding dimension = 100, L = 3, learning rate = 0.001, dropout = 0.3, the relationship weight k in the FB15K dataset was 0.2, and the k value in the FB15K-237 dataset was 0.3.

4.3. Main Result

We used the indicators mentioned earlier to provide the results, and Table 2 reports the results of our model on the FB15K dataset, as well as the comparison results with some baseline models.

Table 2. Results of link prediction on FB15k datasets.

	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	-	0.463	0.749	0.578	0.297
ComplEx	-	0.692	0.884	0.759	0.599
DistMult	42	0.798	0.893	-	-
ConvE	51	0.657	0.831	0.723	0.558
RotatE	40	0.797	0.884	0.830	0.746
RGCN	-	0.696	0.842	0.760	0.601
KTAT	**38**	**0.808**	**0.897**	**0.839**	**0.766**

In Table 2, all baseline model data except for RGCN are from Sun et al. [29], and RGCN data are from Schlichtkrull et al. [18], where the data that perform best in this evaluation indicator are highlighted in bold. According to the data presented in the table, we can see that our knowledge representation model shows stronger performance on the FB15K dataset when conducting link prediction experiments. Specifically, our model outperforms the baseline models in all indicators, and there are slight improvements in the MR, MRR, and Hit@N metrics compared to the second-ranked model. This result demonstrates the effectiveness of our model, and shows the higher robustness in a complex network structure. In addition, it demonstrates that attention mechanism and entity type information could improve the accuracy of predictions of the relevance between entities.

The data sources of the baseline model in Table 3 are the same as those in Table 2 due to the removal of inverse relationships in the FB15K-237 model, and the performance of the knowledge representation model in link prediction tasks is more challenging. From the experimental results in Table 3, it can be found that our model outperforms the baseline model in all indicators except for MR in the FB15K-237 dataset, and its performance in MR indicators is only second to the RotatE model. Specifically, the KTAT model showed a 47.3% improvement in MRR metrics compared to the second-ranked model. Knowledge representation methods based on neural networks typically have better representation capabilities in graph structured data. The experiments show that fusing the graph attention mechanism and node type information can improve the performance of the model, which has certain advantages compared to baseline models.

Table 3. Results of link prediction on FB15k-237 datasets.

	MR	**MRR**	**Hits@10**	**Hits@3**	**Hits@1**
TransE	357	0.294	0.465	-	-
ComplEx	339	0.247	0.428	0.275	0.158
DistMult	254	0.241	0.419	0.263	0.155
ConvE	244	0.325	0.501	0.356	0.237
RotatE	**177**	0.338	0.533	0.375	0.241
RGCN	600	0.164	0.30	0.181	0.10
KTAT	182	**0.498**	**0.625**	**0.424**	**0.355**

4.4. Entity Description Information Fusion

Our model aggregates neighborhood information based on the graph attention mechanism, and the graph attention method is more sensitive to initial embedding. In order to improve the stability of the model, we extracted text description information as initialization vector representation. We conducted experiments to compare the impact of random initialization and text description initialization on model performance, then we conducted three sets of experiments to analyze the impact of fused text description information on model performance by changing the initial embedding method. We call the model that randomly initializes entity vectors KTAT-R. For the groups that integrate entity description information, two methods were used; the first method embeds the words of the entity description statement into the vector for average operation, and reduces the dimensionality to a fixed length vector through pooling, called KTAT-A. The second method concatenates the word vectors in the description statement into a matrix, and using the method mentioned earlier, we use two-layer convolution to treat the vector obtained from the convolution as the initial embedding of the entity, which is called KTAT-C. Our experimental results on the FB15K-237 dataset are shown in Figure 3.

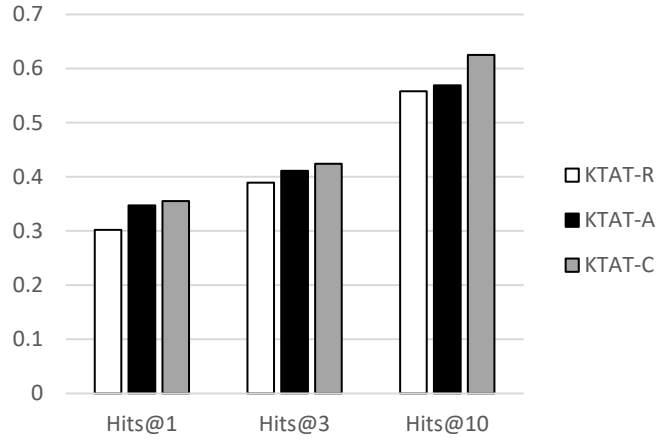

Figure 3. Result of different embedding methods in the FB15K-237 dataset on Hits@N.

From the graph, we find that using text description information to initialize the entity embedding vector can achieve better results than using random initialization embedding in Hit@1, Hit@3, and Hit@10, with the KTAT-C method outperforming the other two methods in all indicators. This indicates that integrating text description information can effectively improve the robustness and stability of the model, and the convolution methods outperform the average methods.

4.5. Relationship Fusion Function

We proposed a relationship fusion $\Phi_{er}(\cdot)$ to aggregate the relationship information in the graph structure message transmission path to the central node, and devised a set of experiments to conduct on this function to verify its effectiveness. We conducted multiple experiments on FB15k and FB15k-237 datasets for different values of hyperparameter k. We analyzed and checked the relationship weight hyperparameter k by sampling it with a fixed step size of 0.1 in the range of 0 to 1. The experimental results obtained using MRR as an evaluation indicator are shown in Figure 4:

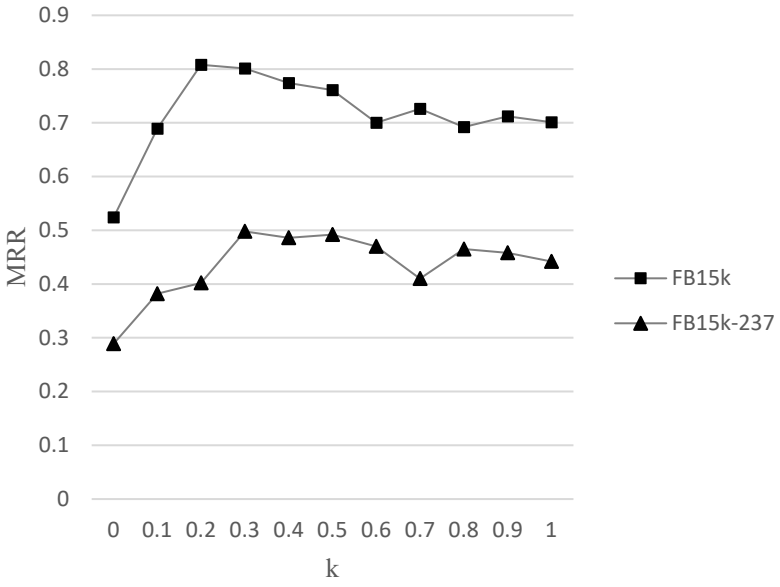

Figure 4. Parameter sensitivity of hyperparameter k.

When k equals 0, it means that the relationship fusion function $\Phi_{er}(\cdot)$ is not used. The experimental results indicate that when k is greater than 0 in both datasets, the experimental results are above the initial value, which means that relationship fusion function $\Phi_{er}(\cdot)$ is

conducive to improvements in the performance of the model. When the value of k is 0.2, our model has the best MRR indicator results on the FB15k dataset. When the value of k is 0.3, our model has the best MRR indicator results on FB15k-237. However, as the value of k gradually increases, the experimental results gradually decrease. The possible reason is that excessive relationship weights may mask neighbor information with lower attention weights according to our analysis.

5. Conclusions

In this paper, we propose a knowledge embedding model based on an attention mechanism and integrating entity attributes. Compared with previous methods of knowledge graph neural networks, this model integrates an attention mechanism into entity type information in heterogeneous knowledge graphs, and uses the attention mechanism to calculate weights to aggregate information features of different weights from neighboring nodes; it can also embed richer semantic features through the types of entities in different roles. We tested our model on the FB15K and FB15K-237 datasets, and the experimental results show that our model outperformed the baseline model in most indicators, and demonstrate the effectiveness on a link prediction task. Our model outperforms all baseline models on the FB15k dataset, and outperforms the baseline model in most indicators on the FB15k-237 dataset; in particular, our model showed a 47.3% improvement in MRR metrics compared to the second-ranked model on the FB15k-237 dataset. The experimental results demonstrate the superiority of the attention mechanism in directed heterogeneous graph structured data, and integrating entity type information contributes to better embedding of the knowledge contained in entities, which can improve the performance of knowledge representation models. However, our model still has certain limitations, including that KTAT aggregates all the neighbors around the center node. Therefore, for nodes with more neighbors, it may lead to the collection of residual noise while aggregating neighborhood information. For future work, a possible research direction is to further improve our model on this point by establishing a new method to aggregate key neighborhood information, block outlier noise, and improve the model's prediction ability for rare nodes. In addition, we will also attempt to apply this model to domain knowledge graphs.

Author Contributions: Conceptualization, Y.L., P.W. and D.Y.; methodology, Y.L.; software, Y.L.; validation, Y.L.; formal analysis, Y.L.; investigation, Y.L.; resources, Y.L.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, P.W.; visualization, Y.L.; supervision, P.W.; funding acquisition, P.W. All authors have read and agreed to the published version of the manuscript.

Funding: Jilin Provincial Science and Technology Innovation Center for Network Database Application (YDZJ202302CXJD027).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krompaβ, D.; Baier, S.; Tresp, V. Type-Constrained Representation Learning in Knowledge Graphs. In Proceedings of the International Semantic Web Conference: The Semantic Web, Bethlehem, PA, USA, 11–15 October 2015; pp. 640–655.
2. Xie, R.; Liu, Z.; Sun, M. Representation learning of knowledge graphs with hierarchical types. In Proceedings of the National Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
3. Lin, Y.; Liu, Z.; Sun, M. Knowledge representation learning with entities, attributes and relations. *IEEE Signal Process. Lett.* **2016**, *23*, 4.
4. Zhang, Z.; Cao, L.; Chen, X.; Tang, W.; Xu, Z.; Meng, Y. Representation Learning of Knowledge Graphs with Entity Attributes. *IEEE Access* **2020**, *8*, 7435–7441.
5. Sun, Z.; Hu, W.; Li, C. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*; Springer: Vienna, Austria, 2017; pp. 628–644.

6. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805v2.
7. Xie, R.; Liu, Z.; Jia, J.; Sun, M. Representation learning of knowledge graphs with entity descriptions. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
8. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-Relational Data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
9. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. *Proc. AAAI Conf. Artif. Intell.* **2014**, *28*, 1. [CrossRef]
10. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge Graph Embedding via Dynamic Mapping Matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 687–696.
11. Dai, S.; Liang, Y.; Liu, S.; Ying, W.; Shao, W. Learning Entity and Relation Embeddings with Entity Description for Knowledge Graph Completion. In Proceedings of the AAAI Conference on Artificial Intelligence, ICAITA 2018, New Orleans, LA, USA, 2–7 February 2018; pp. 194–197.
12. Fan, M.; Zhou, Q.; Chang, E.; Zheng, F. Transition-Based Knowledge Graph Embedding with Relational Mapping Properties. In Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing, Phuket, Thailand, 12–14 December 2014; pp. 328–337.
13. Feng, J.; Zhou, M.; Hao, Y.; Huang, M.; Zhu, X. Knowledge Graph Embedding by Flexible Translation. In Proceedings of the Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning, Cape Town, South Africa, 25–29 April 2016.
14. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online Learning of Social Representations. In Proceedings of the KDD'14: 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014.
15. Tomas, M.; Kai, C.; Greg, C.; Jeffrey, D. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
16. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. In Proceedings of the KDD'16: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
17. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
18. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In *European Semantic Web Conference*; Springer: Cham, Switzerland, 2018; pp. 593–607.
19. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive Representation Learning on Large Graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1025–1035.
20. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *Stat* **2017**, *1050*, 20.
21. Li, Z.; Liu, H.; Zhang, Z.; Liu, T.; Xiong, N. Learning knowledge graph embedding with heterogeneous relation attention networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3961–3973. [CrossRef] [PubMed]
22. Zhang, Z.; Zhuang, F.; Zhu, H.; Shi, Z.; Xiong, H.; He, Q. Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 9612–9619. [CrossRef]
23. Munne, R.; Ryutaro, I. Entity alignment via summary and attribute embeddings. *Log. J. IGPL* **2022**, *31*, 314–324. [CrossRef]
24. Cheng, Y.; Liu, Z.; Zhao, D.; Sun, M.; Chang, E. Network Representation Learning with Rich Text Information. In Proceedings of the International Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
25. Tu, C.; Liu, H.; Liu, Z.; Sun, M. Cane: Context-aware network embedding for relation modeling. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1722–1731.
26. Guo, S.; Wang, Q.; Wang, B.; Wang, L.; Guo, L. SSE: Semantically smooth embedding for knowledge graphs. *IEEE Trans. Knowl. Data Eng.* **2016**, *29*, 884–897. [CrossRef]
27. Ma, S.; Ding, J.; Jia, W.; Wang, K.; Guo, M. Transt: Type-based multiple embedding representations for knowledge graph completion. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, North Macedonia, 18–22 September 2017; Volume 10, pp. 717–733.
28. Toutanova, K.; Chen, D. Observed versus latent features for knowledge base and text inference. In Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Beijing, China, 31 July 2015; pp. 57–66.
29. Sun, Z.; Deng, Z.; Nie, J.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *arXiv* **2019**, arXiv:1902.10197v1.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.