



A Novel Encoder-Decoder Knowledge Graph Completion Model for Robot Brain

Yichen Song, Aiping Li*, Hongkui Tu, Kai Chen and Chenchen Li

College of Computer, National University of Defence Technology, Changsha, China

With the rapid development of artificial intelligence, Cybernetics, and other High-tech subject technology, robots have been made and used in increasing fields. And studies on robots have attracted growing research interests from different communities. The knowledge graph can act as the brain of a robot and provide intelligence, to support the interaction between the robot and the human beings. Although the large-scale knowledge graphs contain a large amount of information, they are still incomplete compared with real-world knowledge. Most existing methods for knowledge graph completion focus on entity representation learning. However, the importance of relation representation learning is ignored, as well as the cross-interaction between entities and relations. In this paper, we propose an encoder-decoder model which embeds the interaction between entities and relations, and adds a gate mechanism to control the attention mechanism. Experimental results show that our method achieves better link prediction performance than state-of-the-art embedding models on two benchmark datasets, WN18RR and FB15k-237.

Keywords: learning-based artificial intelligence, robot intelligence, human-robot interaction, knowledge graph reasoning and completion, knowledge graph embedding

OPEN ACCESS

Edited by:

Zhaoquan Gu,
Guangzhou University, China

Reviewed by:

Xiang Lin,
Shanghai Jiao Tong University, China

Li Xiaoyong,
Beijing University of Posts and
Telecommunications (BUPT), China

Jinqiao Shi,
Beijing University of Posts and
Telecommunications (BUPT), China

*Correspondence:

Aiping Li
13017395458@163.com

Received: 01 March 2021

Accepted: 12 April 2021

Published: 11 May 2021

Citation:

Song Y, Li A, Tu H, Chen K and Li C
(2021) A Novel Encoder-Decoder
Knowledge Graph Completion Model
for Robot Brain.
Front. Neurobot. 15:674428.
doi: 10.3389/fnbot.2021.674428

1. INTRODUCTION

With the development of science and technology, significant progress has been achieved in robotics that the types and application fields of robots are constantly enriched. These robots have played key roles in reducing tedious work. They provide optimal user service and improve the convenience of life. The popularity of various kinds of robots is an inevitable trend.

The emergence of learning intelligent social robots means that robots have truly begun to play roles in people's daily lives, such as pepper and buddy. There are some typical applications, such as greeting conversations, question responses, interest recommendations, and risk management (Gu et al., 2021). Huge information is need at the backend of these services. However, the traditional search engine will be affected by the combination of information, resulting in an increase in search volume and a decrease in accuracy. Knowledge with unique meaning and with the goal of solving practical problems can avoid this problem well.

The "Robot Brian" is taken the same as the brain for humans, which stores and infers knowledge to support other behaviors. Knowledge base is usually used to work as the brain of an intelligent robot. Unlike general applications that implicitly encode information in programs, it can explicitly express the corresponding knowledge of actual problems. Providing continuous knowledge support for robots through the knowledge base is equivalent to injecting "thought" into the robots to realize real intelligence true intelligence. Knowledge base construction is a core configuration for

intelligent robots. Without a knowledge base, a robot cannot answer any questions. The richer the knowledge base, the more intelligent the robot will have when interacting with users.

A large amount of research work in knowledge representing, web data mining, natural language processing and other fields are dedicated to acquiring large-scale knowledge (Jia et al., 2021), providing rich knowledge bases for building the intelligent brain of robots. In order to facilitate computer processing and understanding, we express the knowledge base in a more formal and concise way, that is, a highly structured knowledge graph composed of triples (e_h, r_k, e_t) . The Knowledge Graphs (KG) not only provides robots with a more human-like representation of the world, but also provides a better way to organize, manage and utilize massive amounts of information.

Although the large-scale knowledge graphs already contain a large amount of entity and relation information, they are still incomplete compared with existing knowledge and newly added knowledge (Zhao et al., 2020). Through knowledge graphs and knowledge self-learning, problems in the knowledge system can be found, and knowledge can be supplemented and enhanced so that the robot's knowledge base can be continuously improved and evolved. There is no end to the optimization and completion of the knowledge base, just as there is no end to human learning, this is research work that needs continuous improvement and development.

In order to alleviate the above problems, researchers have proposed a knowledge graph embedding method, which predicts missing links based on existing facts so as to expand the knowledge base. Its purpose is to learn low-dimensional vector representations of all entities and relationships, so as to simplify operations while the original structured information of the knowledge graph is retained. These knowledge graph embedding methods are widely divided into translation models (Bordes et al., 2013; Ji et al., 2015; Lin et al., 2015), semantic matching model (Nickel et al., 2011, 2016; Yang et al., 2015; Trouillon et al., 2016), and neural network models (Dettmers et al., 2018; Shang et al., 2019; Vashishth et al., 2020). The related work will be introduced in detail in section 2.

Compared with neural network models, the other types of models are all shallow models, which leads to problems with poor expressiveness. Therefore, more and more complex and deeper models, which have better expressive performance and have achieved competitive success in modeling knowledge graphs, have been proposed in recent years. But these existing models, such as Dettmers et al. (2018), Nguyen et al. (2018), Shang et al. (2019), and Vashishth et al. (2020), are more focused on entity representation learning, and the importance of relation representation learning are ignored, let alone the cross-interaction between entities and relations. The interaction between entities and relations plays an important role in knowledge graph representation learning that the entities and relations in the knowledge graph will influence each other and influence the prediction of new triples as they do in the real world.

In this paper, a method of knowledge reasoning and completion based on neural networks on the knowledge graph is designed for robots to simulate the reaction and learning process of human brains. Our model adopts the encoder-decoder model. The encoder model improved the KBGAT model with a gate

mechanism to control the attention mechanism and use entity embeddings to update relation embeddings. The decoder model uses Conv-TransE and Conv-TransR to achieve state-of-the-art efforts. This method can enable the robot to quickly search for information, predict answers, and complete knowledge from the knowledge base, to better understand user intent and interact with users more intelligently.

2. RELATED WORK

In this section, we mainly introduce the work related to our Large-scale Knowledge Graph reasoning and completion methods for robots. As one of the research hotspots, Large-scale Knowledge Graph reasoning and completion has attracted extensive attention from academia and industry. Thus, many different types of methods are born, such as the translation model, the bilinear model, the hyperbolic geometry model, the neural network model, the rotate model, and so on. Among these different kinds of methods, the knowledge graph Embedding method is the closest to human expression, which can be regarded as languages for computers and machines like robots. The knowledge graph Embedding method generally includes the following types of models: (i) translation models; (ii) models based on semantic matching; (iii) models based on neural networks; (iv) models with additional information. We will mainly introduce the work related to translation models and neural network based models related to our work in the following.

The translation model represented by TransE (Bordes et al., 2013) uses a simple vector form to represent the entities and relations in the knowledge graph. TransE (Bordes et al., 2013) regards relation as the conversion from the head entity e_h to the tail entity e_t , and uses $e_h + r_k = e_t$ to determine whether the given triplet is correct. In order to make up for the defect that TransE can only handle the 1-1 relation, TransH (Wang et al., 2014), TransD (Ji et al., 2015), TransR (Lin et al., 2015), and other models have increased the ability to handle multiple relations and semantics and enhanced the knowledge embedding model. It shows that entities and relations can also be embedded in other spaces besides real number space. TransG (Xiao et al., 2015) introduces Gaussian distribution to solve the problem of multi-relational semantics to capture the uncertainty of entities and relations. TorusE (Ebisu and Ichise, 2018) is the first model to embed objects outside the space of real or complex numbers and select a torus (compact Lie group) as the embedding space.

Neural network-based embedding models have received extensive attention in recent years. These methods include embedding models based on convolutional neural networks (CNN) and graph convolution networks (GCN). For example, convE (Dettmers et al., 2018), ConvKB (Nguyen et al., 2018), and InteractE (Vashishth et al., 2020) are both relational prediction models based on convolutional neural networks. ConvE (Dettmers et al., 2018) stacks the embeddings of head entity and relation into a 2-dimensional matrix, and performs convolution operation to extract features with fewer parameters and faster calculations. IntercatE (Vashishth et al., 2020) increases the expressive power of ConvE and expands the interaction between entities and relations through three

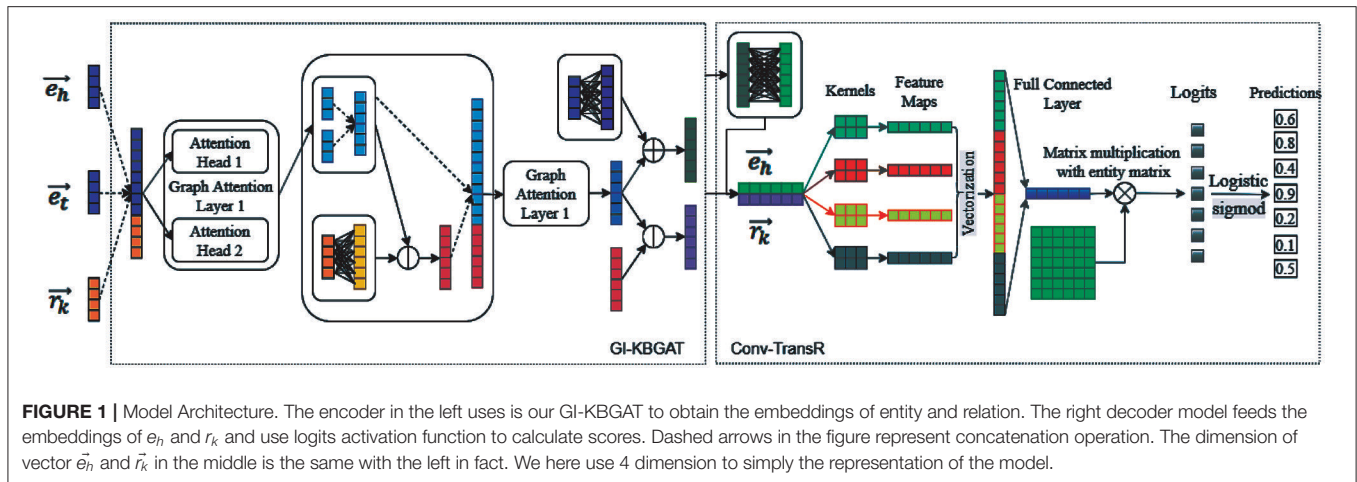


FIGURE 1 | Model Architecture. The encoder in the left uses is our GI-KBGAT to obtain the embeddings of entity and relation. The right decoder model feeds the embeddings of e_h and r_k and use logits activation function to calculate scores. Dashed arrows in the figure represent concatenation operation. The dimension of vector e_h and r_k in the middle is the same with the left in fact. We here use 4 dimension to simply the representation of the model.

key ideas—feature permutation, feature reshaping, and circular convolution. ConvKB (Nguyen et al., 2018) represents each triple as a 3-column matrix where each column vector represents a triple element and feeds this matrix to a 1D convolution layer to generalize transitional characteristics in transition-based embedding models.

Graph convolution networks have made great progress in improving the efficiency of node representation in the graph, and it is also applied in the knowledge graph by researchers. Graph convolution network (GCN) (Kipf and Welling, 2017) gathers information for node(entity) from its neighbors with equal importance. Velickovic et al. (2018) introduce a graph attention network (GAT) to learn to assign varying levels of importance to node(entity) in every neighbor. However, these models are unsuitable for KGs, since they ignore that edges (triples) play different roles depending on the relation they are associated with in KGs. SACN (Shang et al., 2019) extends the classic GCN to a weighted graph convolutional network (WGCN) as an encoder, and uses a convolution model Conv-TransE as a decoder to construct an end-to-end model. WGCN weighs the different types of relations differently when aggregating multiple single-relation graphs into a multi-relation graph and the weights are adaptively learned during the training of the network. But WGCN inherits GCN’s shortcomings in that it treats the same relation type for different entities of the same weight. Nathani et al. (2019) extends classic GAT to KBGAT by incorporating relation and neighboring node features in the attention mechanism and uses KBGAT as encoder and ConvKB (Nguyen et al., 2018) as a decoder.

The above-mentioned models have achieved good performance in knowledge graph embedding for knowledge graph reasoning and completion. However, as far as we know, few works consider the cross-interaction of entities and relations when designing models. Our proposed model uses a variant of the graph attention network (GAT) as the encoder and uses variants of ConvE [Conv-TransE (Shang et al., 2019), Conv-TransR] as decoder, to achieve the simultaneous capture of entity-to-relation and relation-to-entity cross-interaction.

3. MODEL

This section begins by introducing some notations and definitions used in the rest of this article. This is followed by an introduction of our encoder model GI-KBGAT, an improved Graph Attention Network for KG, which considers gate mechanism on multi-head attention and interaction between entities and relations to generate embeddings. Finally, we describe our decoder network based on Conv-TransE (Conv-TransR). The architecture of our model is as shown in **Figure 1**.

3.1. Notations and Definitions

The knowledge graph is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E} = \{e_1, e_2, \dots, e_N\}$ and $\mathcal{R} = \{r_1, r_2, \dots, r_K\}$ represent the set of entities (nodes) and relations. N is number of entities and K is number of relations. \mathcal{T} denotes the triples (edges) of the form $t_{htk} = (e_h, r_k, e_t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where e_h is head entity, e_t is tail entity, and r_k is the relation between head and tail entity. In particular, entity e_h and e_t in this paper refer to the head entity and the tail entity, respectively, while other entities with subscripts, such as entity e_i are not specified. **Table 1** explains the notations that will be used in the rest of this article.

3.2. Encoder: GI-KBGAT

As shown in section 2, most existing models ignore the cross-interaction of entities and relations. They only use relations to updating entities, but ignore the effects of entities on relations. We improve the KBGAT (Nathani et al., 2019) by modifying the update process of embeddings of entities and relations to consider the interaction between entities and relations in the update process, and adding a gate mechanism to the attention mechanism for control.

Our model uses the initial embeddings of entities and relations as input, and the following layers use the embeddings obtained from its previous layer as input. The same as Nathani et al. (2019)’s GAT model, in order to learn the embeddings of entity e_i , we aggregate features of triples associated with it. The triple’s embedding is learned by performing a linear transformation over the concatenation of entity and relation vectors corresponding to

TABLE 1 | Some of the notations and explanations used in this paper.

| Notation | Explanation | Notation | Explanation |
|--------------------------|--|-----------------|---|
| \mathbf{E}, \mathbf{R} | Embedding matrices for entities and relations | $[\parallel]$ | Concatenation of vectors |
| \vec{e}_i, \vec{r}_k | Embedding vectors of entity e_i and relation r_k | $ \mathcal{S} $ | Number of elements in set \mathcal{S} |
| \mathcal{N}_i | Neighbor sets of entity e_i | ϕ | LeakyReLU function |
| \mathcal{R}_i | Relation sets connecting entities e_i and e_j | σ | Activate function |
| \mathcal{P}_k | Entity pair (e_h, e_t) sets with relation r_k | ψ | ReLU function |

it, as shown in Equation (1).

$$\vec{t}_{htk} = \mathbf{W}_t[\vec{e}_h \parallel \vec{e}_t \parallel \vec{r}_k] \quad (1)$$

Where \vec{t}_{htk} , \vec{e}_h , \vec{e}_t , and \vec{r}_k represent the embeddings of triple t_{htk} , entity e_h , e_t , and relation r_k , respectively, \mathbf{W}_t denotes the linear transformation matrix. To measure the importance of each triple t_{htk} for entity e_h , the LeakyRelu non-linearity activation function ϕ is used to get the absolute attention value, the activate vector is defined as \vec{b} , and the softmax function is used to obtain the relative attention values α_{htk} , as shown in Equation (2).

$$\alpha_{htk} = \frac{\exp(\phi(\vec{b}t_{htk}))}{\sum_{n \in \mathcal{N}_h} \sum_{r \in \mathcal{R}_{hn}} \exp(\phi(\vec{b}t_{hnr}))} \quad (2)$$

Then, the new embedding of entity e_h is obtained by aggregating the features of the triples associated with e_h through weighted by their attention values. As shown in Equation (3), attention values are used to calculate the linear combination of triples(neighbor) features, and the embedding is obtained with a activate function σ .

$$\vec{e}_h = \sigma\left(\sum_{t \in \mathcal{N}_h} \sum_{k \in \mathcal{R}_{ht}} \alpha_{htk} \vec{t}_{htk}\right) \quad (3)$$

To stabilize the learning process and encapsulate more information, our encoder also uses a gated multi-head attention mechanism inspired by Vaswani et al. (2017), Velickovic et al. (2018), and Zhang et al. (2018). Considering M independent attention heads, M embeddings for an entity are obtained. For example, the embedding of entity e_h calculated by the m -th attention head is represented as \vec{e}_h^m . These embeddings of an entity are concatenated with independent gate value g_h^m except the last layer (for which we use the mean pooling). The final entity embedding update equation is as follows:

$$\vec{e}_h = \sigma\left(\left[\parallel_{m=1}^M (g_h^m \cdot \sigma\left(\sum_{t \in \mathcal{N}_h} \sum_{k \in \mathcal{R}_{ht}} \alpha_{htk} \vec{t}_{htk}\right)^m\right)\right]\right) \quad (4)$$

For the relation update, we propose an update mechanism that uses the same projection operation as the TransR model (Lin et al., 2015). Similar to GAT, TransR model holds that an entity is a complex of various attributes, and different relations focus

on different attributes of the entity. TransR uses the projection matrix \mathbf{M}_r to project the head entity e_h and tail entity e_t into the corresponding relation space, and defines the score function as $f_r(e_h, e_t) = \|\vec{e}_h \mathbf{M}_r + \vec{r} - \vec{e}_t \mathbf{M}_r\|_2^2$. Inspired by TransR, we project the head entity e_h and the tail entity e_t of a triple into the relation space with a projection matrix \mathbf{W}_r , and update their relation r_k as Equation (5).

$$\vec{r}_k = \frac{1}{|\mathcal{P}_k|} \sum_{(h,t) \in \mathcal{P}_k} (\vec{e}_t - \vec{e}_h) \mathbf{W}_r \quad (5)$$

In order not to lose the initial embeddings information during training, our model design a gated mechanism to aggregate the initial embeddings and the updated embeddings with learnable gate values. The equation is as shown in Equation (6).

$$\begin{aligned} \mathbf{E} &= g_{ei} \mathbf{E}_{initial} \mathbf{W}_{te} + g_{eu} \mathbf{E}_{update} \\ \mathbf{R} &= g_{ri} \mathbf{R}_{initial} \mathbf{W}_{tr} + g_{ru} \mathbf{R}_{update} \end{aligned} \quad (6)$$

Where \mathbf{W}_{te} and \mathbf{W}_{tr} are the linearly transform matrices for the initial embeddings of entity $\mathbf{E}_{initial}$ and relation $\mathbf{R}_{initial}$, g_{ei} , and g_{ri} are the memory gate for the initial embeddings of entity $\mathbf{E}_{initial}$ and relation $\mathbf{R}_{initial}$, g_{eu} , and g_{ru} are the update gate for the updated embeddings of entity \mathbf{E}_{update} and relation \mathbf{R}_{update} obtained by Equations (4) and (5), respectively. The scoring function for the GI-KBGAT method is defined as follows:

$$f(\Omega) = \sum_{t_{htk} \in \mathcal{S}} \sum_{t'_{htk} \in \mathcal{S}'} (\|\vec{e}_h + \vec{r}_k - \vec{e}_t\|_1 - \|\vec{e}_h + \vec{r}'_k - \vec{e}'_t\|_1) \quad (7)$$

Where \mathcal{S} and \mathcal{S}' denotes the set of valid triples [$t_{htk} = (e_h, r_k, e_t)$] and invalid triples [$t'_{htk} = (e'_h, r'_k, e'_t)$], respectively, $\|\cdot\|_1$ means L1-norm dissimilarity.

3.3. Decoder

The convolutional structure is used as the base model of our decoder, which transforms the embedding vector to another space and possesses powerful feature extraction ability and good parameter efficiency. The decoder takes the embeddings of entity and relation trained from the encoder as input. We test both Conv-TransE (Shang et al., 2019), and Conv-TransR, which keeps the translational property of TransR ($\vec{e}_h \mathbf{W} + \vec{r}_k \approx \vec{e}_t \mathbf{W}$) with 1D convolution inspired by Conv-TransE to be consistent with encoder, as decoder as shown in **Figure 1**.

The only difference between Conv-TransR and Conv-TransE is that the Conv-TransR model has one more project matrix for entities than Conv-TransE. Following shows the model of Conv-TransR: Conv-TransR uses matrix \mathbf{W} to project the entity e_h into its corresponding relation r_k 's space, the result is $\vec{e}_h \mathbf{W}$. This result is then stacked with its corresponding relation embedding \vec{r}_k to get $[\vec{e}_h \mathbf{W}, \vec{r}_k]$ as the input of convolutional network. The convolutional network uses different filters(kernels) $\omega \in \mathbb{R}^{2 \times F}$ ($F \in \{1, 2, 3, \dots\}$) to generate different feature maps as Conv-TransE. The scoring function for the Conv-TransR method is defined as below:

$$g(e_h, r_k, e_t) = \tau(\psi([\psi([\vec{e}_h \mathbf{W}, \vec{r}_k] \otimes \omega)] \mathbf{W}_c) \vec{e}_t) \quad (8)$$

Where \otimes represents a 1D convolution operation, $[\parallel]$ denotes vector concatenation which concatenates features output from convolution with different filters ω , W_c is a learnable weight matrix for linear transformation to projected the concatenation embedding into the tail entity e_t space, ψ is chosen to be a ReLU non-linear function, then the calculated embedding is matched to tail entity e_t by an appropriate distance metric, and the logistic sigmoid function τ is used for scoring finally.

4. EXPERIMENTS AND RESULTS

4.1. Datasets

Through continuous learning, the data scale of intelligent robots will only increase. Therefore, when evaluating our proposed method, we ignore the small datasets and chose two large datasets WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova et al., 2015) as the benchmark datasets. WN18RR and FB15k-237 are improved versions of two common datasets WN18 and FB15k (Bordes et al., 2013) derived from WordNet and freebase, respectively, in which all inverse relations have been deleted to prevent direct inference of test triples by reversing training triples. Table 2 provides statistics of them.

4.2. Training Settings

We follow a two-step training procedure that, we first train our GI-KBGAT to encode information about the graph entities and relations and then train decoder model Conv-TransR to perform the link prediction task. For encoder training, we use the margin ranking loss, use Adam to optimize all the parameters with

the initial learning rate set at 0.001, set the entity and relation embedding dimension of the last layer to 200, and set the other hyper-parameters for each dataset to be the same as KBGAT (Nathani et al., 2019). For decoder training, we use the standard binary cross-entropy loss with label smoothing, set the size and number of the kernel to 9 and 200, respectively, and set the other hyper-parameters for each dataset to be the same as InteractE (Vashishth et al., 2020).

4.3. Evaluation Protocol

Following the previous work, we use the filtered setting (Bordes et al., 2013) that all valid triples are filtered out from the candidate set while evaluating test triples. The performance is reported on the standard evaluation metrics: Mean Reciprocal Rank (MRR) and the proportion of correct entities ranked in the top 1, 3, and 10 (Hits@1, Hits@3, Hits@10).

4.4. Results and Analysis

Table 3 presents the experimental results of our methods and several baseline methods on FB15K-237 and WN18RR test sets. In which all values are presented in percentage. Among these baseline methods, the methods in the first box, namely TransE (Bordes et al., 2013), ConvE (Dettmers et al., 2018), ConvKB (Nguyen et al., 2018), Conv-TransE (Shang et al., 2019), and InteractE (Vashishth et al., 2020), have their results taken from the original paper and can be resumed to acceptable results. We compared our methods with these methods in Table 3 to label the best score in bold.

TABLE 2 | Statistics of the experimental datasets.

| Dataset | # Entities \mathcal{E} | # Relations \mathcal{R} | # Edges \mathcal{T} | | | | Mean in-degree | Median in-degree |
|-----------|-------------------------------|--------------------------------|-------------------------|------------|---------|---------|-------------------|---------------------|
| | | | Training | Validation | Testing | Total | | |
| WN18RR | 40,943 | 11 | 86,835 | 3,034 | 3,134 | 93,003 | 2.12 | 1 |
| FB15k-237 | 14,541 | 237 | 272,115 | 17,535 | 20,466 | 310,116 | 18.71 | 8 |

TABLE 3 | Experimental results on FB15K-237 and WN18RR test sets.

| Models | FB15k-237 | | | | WN18RR | | | |
|----------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 |
| TransE Bordes et al., 2013 | 29.4 | – | – | 46.5 | 22.6 | – | – | 50.1 |
| ConvE Dettmers et al., 2018 | 32.5 | 23.7 | 35.6 | 50.1 | 43 | 40 | 44 | 52 |
| ConvKB Nguyen et al., 2018 | 39.6 | – | – | 51.7 | 24.8 | – | – | 52.5 |
| Conv-TransE Shang et al., 2019 | 33 | 24 | 37 | 51 | 46 | 43 | 47 | 52 |
| InteractE Vashishth et al., 2020 | 35.4 | 26.3 | – | 53.5 | 46.3 | 43.0 | – | 52.8 |
| SACN Shang et al., 2019 | 35 | 26 | 39 | 54 | 47 | 43 | 48 | 54 |
| KBGAT Nathani et al., 2019 | 20.5 | 11.4 | 22.8 | 39.6 | 40.4 | 32.2 | 44.8 | 55.4 |
| Our encoder model (+ ConvTransE) | 35.5 | 26.3 | 39.1 | 53.8 | 45.9 | 42.8 | 46.7 | 52.5 |
| Our encoder model (+ ConvTransR) | 33.9 | 24.4 | 37.6 | 52.8 | 46.6 | 43.4 | 47.9 | 52.9 |
| Our encoder model (+ InteractE) | 35.5 | 26.2 | 39.2 | 54.1 | 46.7 | 43.5 | 48.1 | 52.9 |

All values are in percentage and the best scores of our model is in bold regardless of the second box (SACN and KBGAT).

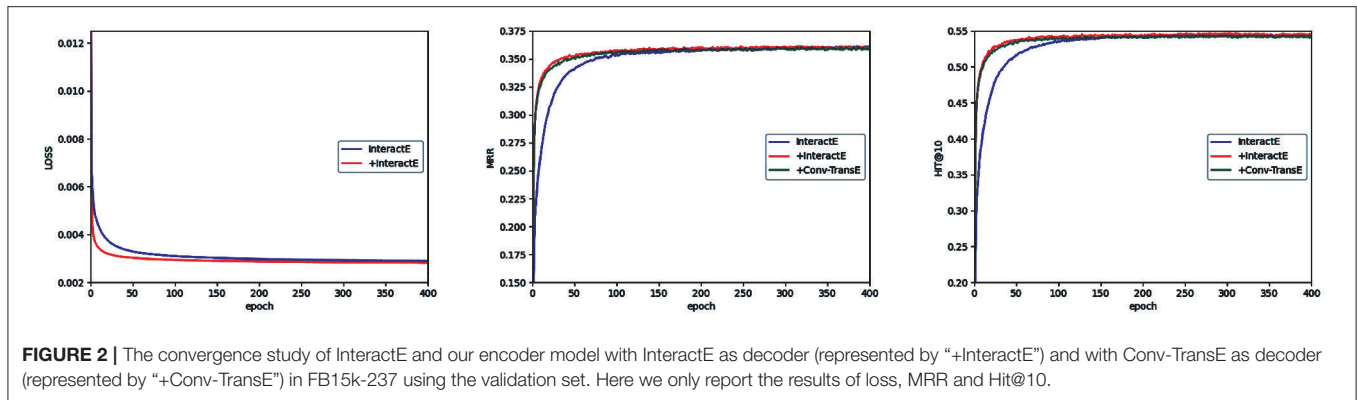


FIGURE 2 | The convergence study of InteractE and our encoder model with InteractE as decoder (represented by “+InteractE”) and with Conv-TransE as decoder (represented by “+Conv-TransE”) in FB15k-237 using the validation set. Here we only report the results of loss, MRR and Hit@10.

TABLE 4 | Ablation experimental results on WN18RR test sets.

| Models | WN18RR | | | |
|-------------------------------------|-------------|-------------|-------------|-------------|
| | MRR | H@1 | H@3 | H@10 |
| KBGAT encoder (+ KBGAT decoder) | 40.4 | 32.2 | 44.8 | 55.4 |
| Our encoder model (+ KBGAT decoder) | 41.1 (+0.7) | 33.1 (+0.9) | 45.5 (+0.7) | 55.9 (+0.5) |
| Our encoder model (+ ConvTransR) | 46.6 | 43.4 | 47.9 | 52.9 |
| –gate | 46.1 (–0.5) | 43.0 (–0.4) | 47.3 (–0.6) | 52.4 (–0.5) |
| –rel | 46.2 (–0.4) | 43.0 (–0.4) | 47.5 (–0.4) | 52.7 (–0.2) |
| –gate – rel | 46.1 (–0.5) | 42.9 (–0.5) | 47.3 (–0.6) | 52.2 (–0.7) |

Since our method is inspired by methods SACN (Shang et al., 2019) and KBGAT (Nathani et al., 2019) that we present their results in the second box. The results of the SACN model are obtained from its corresponding paper, but this model requires a large GPU to train and these results can not be reproduced with the authors’ code. KBGAT has test data leakage in its original implementation that the results in its paper are not credible. In our experiment results table, we fix the problem and show the correct results of the model.

We first compare our model use Conv-TransE as the decoder with the Conv-TransE model. Our model performs better than Conv-TransE on both datasets. Especially in the FB15K-237 dataset, our model improves upon Conv-TransE’s MRR by a margin of 7.6%, Hits@1 of 9.6%, Hits@3 of 5.7%, and Hits@10 of 5.5%. In the WN18RR dataset, our model improves upon Conv-TransE’s Hits@10 by a margin of 1.0%. Under the same accuracy, our model achieves the same performance on the other metrics compared with Conv-TransE.

Second, we compare our model use InteractE as the decoder with the InteractE model to better prove the effectiveness of our encoder. As shown in the Table 3, compared with the original model, most metrics of InteractE have been improved after our encoder model is added. For example, our model with InteractE improves upon InteractE’s MRR by a margin of 0.3% and Hits@10 of 1.1% in the FB15K-237 dataset.

Third, we compare our model with the other baseline models. In the FB15K-237 dataset, our model with Conv-TransE as decoder achieves the best performance in Hit@3 and Hit@10, and tied for the best in Hit@1. In the WN18RR dataset, our model

with Conv-TransR as decoder achieves the best performance in all metrics. Meanwhile, these two models both can achieve the top three effects on the other datasets. In conclusion, our model can achieve the best results on both datasets FB15K-237 and WN18RR.

Figure 2 shows the convergence of the three models: InteractE, our encoder model with InteractE as the decoder, and Conv-TransE as the decoder. Because Conv-TransE uses a different loss function, we do not put its loss result for comparison. We can see that our models (the red line and green line) are always better than InteractE (the blue line) under MRR and Hit@10. And our models converge faster than InteractE.

4.5. Ablation Experiment

In order to prove the validity of our model, we do some ablation experiments on WN18RR dataset to show the influences of different parts of our model. The results of the ablation experiments are shown in Table 4. Compared with the KBGAT model, our improved encoder with the KBGAT decoder (convKB) can achieve better performance as shown in the first box. After changing the decoder methods, our method performs more superior as in the second box. Combined with Table 3, the decoders we use are both better than the ConvKB which is used by KBGAT, which shows the effectiveness of our decoder chosen.

To better show the influences of our innovation in KBGAT, we also test the influence of different parts in our encoder. We separately remove the gate mechanism, the interaction mechanism, and both of them to see the impact on results. The results are shown in the second box of Table 4. The gate

mechanism and the interaction mechanism both perform a similar influence on the encoder model. And the best result can be achieved by combining them in the KBGAT model as the encoder model.

For these results, we conclude that our encoder-decoder model can better the expressive performance of entity embeddings and relation embeddings, and can achieve competitive success in modeling knowledge graphs. Since the hyper-parameters of our model for each dataset are set to the same as the existing methods and no parameter tuning is performed to obtain the best performance, we believe that the performance of our model can still be improved with parameter tuning.

5. CONCLUSION

In this paper, we propose a novel approach for knowledge graph relation prediction, which can be used in intent understanding in human-robot interaction and in robots' knowledge graph completion. Our methods can work well in large-scale knowledge graphs and can be extended to learn embeddings for various applications of robots, such as dialog generation and question answering.

In the future, we intend to extend our method to work as an end-to-end work and consider the attribute information and temporal information into our model to improve the ability to handle complex knowledge graphs. And we also intend to test

our work in a real robot's "brain" to test the ability of our model in actual work.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found at: two datasets used: WN18RR and FB15k-237. Download from <https://github.com/TimDettmers/ConvE>.

AUTHOR CONTRIBUTIONS

YS, KC, and CL contributed to the conception and design of the study. KC wrote the sections of the code. CL wrote the parts of the first draft of the manuscript. YS organized the experiments and the manuscript. AL and HT directed the whole work and revised the manuscript. All authors approved the submitted version.

FUNDING

This work described in this paper was partially supported by the National Key R&D Program of China (Nos. 2016QY03D0603, 2017YFB0802204, 2019QY1406, and 2017YFB0803303), the Key R&D Program of Guangdong Province (No. 2019B010136003), and the National Natural Science Foundation of China (Nos. 61732004, 61732022, 62072131, and 61672020).

REFERENCES

- Bordes, A., Usunier, N., García-Durán, A., Weston, J., and Yakhnenko, O. (2013). "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5–8, 2013*, eds C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger (Lake Tahoe, NV), United States, 2787–2795.
- Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). "Convolutional 2D knowledge graph embeddings," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), The 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), February 2–7, 2018*, eds S. A. McIlraith and K. Q. Weinberger (New Orleans, LA: AAAI Press), 1811–1818.
- Ebisu, T., and Ichise, R. (2018). "Toruse: knowledge graph embedding on a lie group," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), The 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), February 2–7, 2018*, eds S. A. McIlraith and K. Q. Weinberger (New Orleans, LA: AAAI Press), 1819–1826.
- Gu, Z., Wang, L., Chen, X., Tang, Y., Wang, X., Du, X., et al. (2021). Epidemic risk assessment by a novel communication station based method. *IEEE Trans. Netw. Sci. Eng.* 1. doi: 10.1109/TNSE.2021.3058762. [Epub ahead of print].
- Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. (2015). "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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26–31, 2015, Volume 1: Long Papers* (Beijing: The Association for Computer Linguistics), 687–696. doi: 10.3115/v1/P15-1067
- Jia, Y., Gu, Z., and Li, A. (2021). *MDATA: A New Knowledge Representation Model*. Springer International Publishing. doi: 10.1007/978-3-030-71590-8
- Kipf, T. N., and Welling, M. (2017). "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, April 24–26, 2017, Conference Track Proceedings* (Toulon: OpenReview.net).
- Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25–30, 2015*, eds B. Bonet and S. Koenig (Austin, TX: AAAI Press), 2181–2187.
- Nathani, D., Chauhan, J., Sharma, C., and Kaul, M. (2019). "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, July 28–August 2, 2019, Volume 1: Long Papers*, eds A. Korhonen, D. R. Traum, and L. Márquez (Florence: Association for Computational Linguistics), 4710–4723. doi: 10.18653/v1/P19-1466
- Nguyen, D. Q., Nguyen, T. D., Nguyen, D. Q., and Phung, D. Q. (2018). "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, June 1–6, 2018, Volume 2 (Short Papers)*, eds M. A. Walker, H. Ji, and A. Stent (New Orleans, LA: Association for Computational Linguistics), 327–333. doi: 10.18653/v1/N18-2053
- Nickel, M., Rosasco, L., and Poggio, T. A. (2016). "Holographic embeddings of knowledge graphs," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016*, eds D. Schuurmans and M. P. Wellman (Phoenix, AZ: AAAI Press), 1955–1961.
- Nickel, M., Tresp, V., and Kriegel, H. (2011). "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, June 28–July 2, 2011*, eds L. Getoor and T. Scheffer (Bellevue, WA: Omnipress), 809–816.

- Shang, C., Tang, Y., Huang, J., Bi, J., He, X., and Zhou, B. (2019). "End-to-end structure-aware convolutional networks for knowledge base completion," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, January 27–February 1, 2019* (Honolulu, HI: AAAI Press), 3060–3067. doi: 10.1609/aaai.v33i01.33013060
- Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. (2015). "Representing text for joint embedding of text and knowledge bases," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, September 17–21, 2015*, eds L. Márquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton (Lisbon: The Association for Computational Linguistics), 1499–1509. doi: 10.18653/v1/D15-1174
- Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). "Complex embeddings for simple link prediction," in *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, June 19–24, 2016, Volume 48 of JMLR Workshop and Conference Proceedings*, eds M. Balcan and K. Q. Weinberger (New York, NY: JMLR.org), 2071–2080.
- Vashishth, S., Sanyal, S., Nitin, V., Agrawal, N., and Talukdar, P. P. (2020). "Interact: improving convolution-based knowledge graph embeddings by increasing feature interactions," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, February 7–12, 2020* (New York, NY: AAAI Press), 3009–3016. doi: 10.1609/aaai.v34i03.5694
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017*, eds I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, et al. (Long Beach, CA), 5998–6008.
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, April 30–May 3, 2018, Conference Track Proceedings* (Vancouver, BC: OpenReview.net).
- Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27–31, 2014*, eds C. E. Brodley and P. Stone (Québec City, QC: AAAI Press), 1112–1119.
- Xiao, H., Huang, M., Hao, Y., and Zhu, X. (2015). Transg: a generative mixture model for knowledge graph embedding. *CoRR* abs/1509.05488.
- Yang, B., Yih, W., He, X., Gao, J., and Deng, L. (2015). "Embedding entities and relations for learning and inference in knowledge bases," in *3rd International Conference on Learning Representations, ICLR 2015, May 7–9, 2015, Conference Track Proceedings*, eds Y. Bengio and Y. LeCun (San Diego, CA).
- Zhang, J., Shi, X., Xie, J., Ma, H., King, I., and Yeung, D. (2018). "Gaan: gated attention networks for learning on large and spatiotemporal graphs," in *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, August 6–10, 2018*, eds A. Globerson and R. Silva (Monterey, CA: AUAI Press), 339–349.
- Zhao, X., Jia, Y., Li, A., Jiang, R., and Song, Y. (2020). Multi-source knowledge fusion: a survey. *World Wide Web* 23, 2567–2592. doi: 10.1007/s11280-020-00811-0

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Song, Li, Tu, Chen and Li. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.