

A Review of Inference Methods Based on Knowledge Graph

Dexiang Zhang¹, Hairong Wang, Yudan Ding
North Minzu University, China

Abstract. With the development of Internet and big data technology, the scale of data is growing exponentially, and these data contain a lot of valuable information. As the most intuitive way of knowledge expression, knowledge map can effectively organize and express data. As an important means of knowledge map completion, knowledge inference aims to deduce new knowledge or identify wrong knowledge based on existing knowledge in the knowledge map. Different from traditional knowledge inference methods, knowledge inference methods based on knowledge graphs are also diversified according to their simple, intuitive, flexible and rich knowledge expression forms. According to the types of reasoning methods, knowledge reasoning methods based on knowledge graph can be divided into single-step reasoning and multi-step reasoning. According to the different methods adopted for each type, each type also includes reasoning based on distributed representation; reasoning based on neural network and mixed reasoning. These methods are summarized in detail, and the future research direction and prospect of knowledge inference based on knowledge map are discussed and prospected.

Keywords. Knowledge mapping, knowledge inference, distributed representation, neural networks, hybrid networks

1. Introduction

A formalized representation method of the knowledge graph is to use fact triples [1] to record various things and their interrelationships. It transforms human knowledge into a form understandable and usable by machines. It can be represented by the "head entity" and "tail entity" in the above triples, and the relationship between things is represented by the "relationship" in the triples [2]. At present, because most of the open knowledge graphs, such as DBpedia [3], Freebase [4], are constructed manually or semi-automatically, these knowledge

The knowledge map is relatively sparse, and the hidden relationships among a large number of entities have not been fully explored. According to statistics in 2014, in the Freebase knowledge base, 71% of people have no exact date of birth, and 75% have no nationality information [5]. The incompleteness of the knowledge graph seriously restricts the development of the field of artificial intelligence. How to complete the knowledge graph has become one of the important issues in the field of knowledge graph research

¹ Corresponding Author, Hairong Wang, North Minzu University, China; E-mail: bmdwhr@163.com. This work has been supported by the Natural science foundation of Ningxia province (No. 2020AAC03218), and the North Minzu University key research project (No.2019KJ26) and the North Minzu University Graduate innovation program (No. S2020-11407-039G)

ch. Knowledge Graph Completion (KGC) [6] technology was born to deal with the above problems. The knowledge graph completion technology adopts the method of knowledge reasoning, using the existing explicit knowledge in the knowledge graph to predict the tacit knowledge that has not been stored in the graph, and gradually complete the knowledge graph [7].

Traditional knowledge reasoning mainly includes logical reasoning and non-logical reasoning. The process of logical reasoning includes strict constraints and reasoning process, while the process of non-logical reasoning is relatively fuzzy. Logical reasoning is mainly divided into two categories according to the different ways of reasoning: deductive reasoning [8] and inductive reasoning [9]. Among them, inductive reasoning includes abductive reasoning [10] and analogical reasoning [11]. In recent years, the knowledge graph has continued to develop. The number of examples in the graph has gradually increased, and the content covered has become more and more extensive, and a large number of logical rules are required. Therefore, whether it is at the instance level or the abstract concept level, traditional reasoning methods are faced with complexity Problems such as difficulty in solving instantiation problems. In the face of situations where traditional reasoning methods cannot meet the needs, on the basis of significant progress in distributed representation and neural network technology, reasoning based on knowledge graphs has evolved its unique reasoning method.

Compared with traditional knowledge reasoning methods, knowledge reasoning based on knowledge graphs is dominated by the knowledge graph itself, not limited to abstract reasoning at the main conceptual level of the ontology, and can have more specific reasoning methods [12]. There are mainly distributed representation-based methods, neural network-based methods and a mixture of multiple methods. Knowledge reasoning technology is affected by the development of knowledge graphs, and has received wide attention as the main means of knowledge graph completion and knowledge graph denoising.

2. Reasoning based on knowledge graph

The reasoning based on the knowledge graph mainly revolves around the reasoning of the relationship, inferring unknown facts or relationships from the facts or relationships that already exist in the knowledge graph [13], generally focusing on the three aspects of the feature information of entities, relationships and graph structure. Figure 1 shows the reasoning of the character relationship graph, using reasoning to get new facts and rules. In general, reasoning oriented to knowledge graphs can mainly assist in reasoning new facts, new relationships, new axioms, and new rules.

Specifically, the knowledge graph usually uses triples containing head entities, relationships, and tail entities to express the attributes of things and the semantic relationships between things. Among them, the entities in the triples are composed of transaction s and attribute values, and the relationship in the triple is composed of attributes and relationships. Knowledge graph completion [14] is actually given any two elements in the triplet, the other missing element is obtained through the inference algorithm. That is, given the head entity and the relationship or the relationship and the tail entity, find the tail entity or head entity of the matching triple. This process is called entity prediction [15]. In the same way, relationship prediction is to find out the relationship of matching triples under the premise of given head entity and tail entity in the triple. Regardless of entity prediction or relationship prediction, it will be transformed into entities or relations

onships that are more likely to form a valid triple under the premise of a given element as the prediction result, and the effectiveness of the entity can be obtained by calculating the score function of a specific hypothesis.

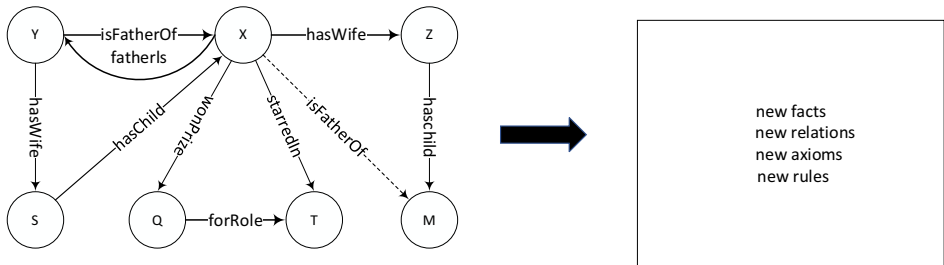


Figure 1 Character diagram reasoning

Since there are thousands of knowledge in the real world, it is very difficult to cover all knowledge, which will lead to the incompleteness of the knowledge map. Therefore, it is very necessary to complete the knowledge map. In the process of completion, Link test is a very typical reasoning task. The triples in the knowledge graph can be obtained through manual definition or text extraction. Due to the limitations of artificial knowledge and the uncertainty of algorithms, there may be conflicting or incorrect information in the knowledge graph. Therefore, inconsistency detection is also an important reasoning task in the knowledge graph. The large amount of information stored in the knowledge graph can provide good knowledge services and provide correct relevant knowledge information for related queries. However, the vagueness of the query and the rich semantic information of the knowledge graph make the query difficult, and knowledge reasoning can help query rewriting. Effectively improve the quality of query results.

By summarizing the reasoning methods based on the knowledge graph, according to the reasoning type, the current reasoning methods are divided into two categories: single-step reasoning and multi-step reasoning. According to the different methods used in each category, further division and explanation of knowledge reasoning methods are made.

3 Single-step reasoning

Single-step reasoning refers to the method of learning and reasoning using the fact tuples stored in the knowledge graph. Its main feature is that it does not consider path characteristics. According to the different methods used, it can be divided into reasoning based on distributed representation, reasoning based on neural network and hybrid reasoning [12].

3.1 Distributed Representation Reasoning

Distributed knowledge representation refers to the transformation of entities and relationships in triples into dense low-dimensional real-valued vectors. Distributed knowledge representation includes two types of vectors: entity vector and relationship

vector, where the entity vector can represent either the head entity or the tail entity. The two entities in the knowledge graph and the relationship between them are represented by triples in the form of (h, r, t), where h is the head entity vector, r is the tail entity vector, and t is the relationship vector. Single-step reasoning based on distributed representation refers to a direct reasoning process based on distributed knowledge representation. Among them, the most typical is representational reasoning based on translation. The idea of this method is inspired by the experimental results of word2vec. Mikolov et al. proposed a word2vec word representation learning model and toolkit [16], on the basis of which they found that there is a phenomenon of translation invariance between the trained word vectors, and found through analogy inference experiments that this translation invariant The phenomenon generally exists in the semantic relationship of words.

TransE: Based on the knowledge of hierarchical knowledge and natural transfer representation of knowledge base, Bordes [17] et al. proposed the first transfer-based representation model TransE. The main idea is: if the triple is established, the head entity vector h The sum of the relation vector r is close to the tail entity vector t , otherwise the three will be far away.

The score function is obtained from the above basic assumptions:

$$f(h, t) = \|r + h - t\|_{L1/L2} \tag{1}$$

Where L_1 or L_2 represents the norm, and the norm is used to measure the distance. The learning process replaces the head entity or the tail entity to get a negative example, similar to a support vector machine, which minimizes the loss function based on Margin. The score of the positive example is at least one Margin higher than the score of the negative example. In reasoning, the candidate entity or relationship with a large score function is the reasoning result. TransE defines the loss function by defining the maximum interval:

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'} [\gamma + d(h + r, t) - d(h' + r', t')] + \tag{2}$$

The $[\gamma + d(h + r, t) - d(h' + r', t')] +$ means $\max(a,x)$; γ is a constant, which means the distance between positive and negative samples.

Compared with the previous model, the TransE model has fewer parameters, low computational complexity, and significantly improved performance. However, for different types of relationships, all entity vectors are represented the same; this leads to the TransE model being effective when solving one-to-one relationships, but not applicable when dealing with complex relationships.

TransR: Both TransE and TransH models put entities and relationships in the same semantic space, which limits the expressive ability of the model. In response to this problem, Lin et al. proposed the TransR model [18], which established entity vectors and relationship vectors in independent entity space and relationship space. First, the entities are projected from the entity space to the corresponding relationship space, and then the relationship between the projected entities is translated. The main idea of the TransR model is: Given a triple (h, r, t), project the head entity and tail entity vectors into the relationship space to get:

$$h_r == M_r h, t_r = M_r t \tag{3}$$

Among them, M_r is the projection matrix from the entity space to the relational space; h is the entity space; t is the relational space.

The potential energy function of the TransR model is:

$$f_r(h, t) = \| h_r + r - t_r \|_{L_n} \tag{4}$$

The TransR algorithm maps entities and relationships into two spaces respectively, and then transfers the entities in the entity space to the relationship space through the transition matrix M_r for vector representation. In the TransR algorithm, the entity vector of each triple is set I_s is a k -dimensional vector, the relationship vector is set to a d -dimensional vector, but k may not be equal to d , and the mapping matrix is a $k \times d$ matrix, so that after transferring from the entity space to the relationship space, each head entity and The tail entities all become vector representations of the relational space. Since the relationships in the graph are usually different, the TransR model learns a unique vector for each relationship. Therefore, the TransR model is not enough to represent all entity pairs under the relationship.

TransD: The TransD model uses two vectors to represent entities and relationships, one is used to represent the meaning of the entity, and the other is used to dynamically construct a mapping matrix [19]. Compared with TransR and CTransR [18], the TransD model not only considers the diversity of relationships but also the diversity of entities. TransD has few parameters and does not require matrix vector multiplication. The main idea of the TransD model is: for a given triple (h, r, t) , its vectors are h, h_p, r, r_p, t, t_p , and the subscript p represents the projection vector, defining two projections .

The matrix maps entities from physical space to relational space:

$$M_{rh} = r_p h_p^t + I^{m \times n}, M_{rt} = r_p t_p^t + I^{m \times n} \quad (5)$$

Among them: $h, h_p, t, t_p \in R^n, r, r_p \in R^n, M_{rh}, M_{rt} \in R_{m \times n}$. Therefore, the mapping matrix is determined by the entity and the relationship. This operation enables the two projection vectors to interact well with each other Function, because each element of them can satisfy each element from another vector. When we initialize each mapping matrix, we will add it to M_{rh} and M_{rt} . Using the mapping matrix, we define the projection vector as follows:

$$h_{\perp} = M_{rh}, t_{\perp} = M_{rt} t \quad (6)$$

Compared with the TransR model, TransD sets up projection matrices for the head entity and tail entity. In addition, notice that there is no matrix-vector multiplication operation after the formula is expanded in TransD. This reduces the computational complexity compared to the TransR model and is more suitable for Calculation of large-scale knowledge graph. In addition to the representation learning methods mentioned so far, there are many other representation learning methods, such as pure neural network methods NTN, ConvE.[20], etc.

From the current analysis of related technologies, the research work of the TransE series of transfer-based representation reasoning methods is relatively comprehensive. However, when modeling, only the constraints of the knowledge graph fact tuple are usually considered, and the deeper combination of semantic information is not considered, and the reasoning ability is limited. The TransE method has the characteristics of simplicity, effectiveness, and high computational efficiency. This has also attracted widespread attention. Many researchers have gone deep into it, and a series of methods have emerged to form the Trans series. However, the TransE series model has been developed for many years, and the development space is trending. Because of saturation, the development potential is small. In addition, the problem-based improvement development route of the Trans series is not suitable for this type of method. More innovative methods, such as methods in the field of mathematics, and modeling spatial distribution are needed.

3.2 Neural Network Reasoning

In single-step reasoning, neural network-based reasoning uses the neural network to directly model the knowledge graph fact tuple, and obtain the vector representation of the fact tuple element for further reasoning. In this method, the entire network constitutes a score function, and the output value of the neural network is used as the score value of the score function. Typical single-step neural network inference models include neural tensor network NTN (neural tensor network), and shared variable neural network model ProjE.

NTN: In order to be able to discover and mine the implicit relationship between different entities in the knowledge graph, Socher et al.[21] proposed a neural tensor network (NeuralTensorNetwork, NTN) method, which uses a bilinear tensor layer to replace the traditional standard linear neural Network layer. By representing each entity in the database as a vector to obtain the facts about the entity and the probability that it belongs to a certain relationship, each relationship is defined by the parameters of the neural tensor network, and the relationship is expressed as a third-order Zhang It can accurately predict other facts under the condition of using the existing database. The advantage of expressing the relationship as a tensor is that each slice can correspond to a different semantic type, that is, a relationship has multiple slices, which can better model the different semantic connections between different entities under the relationship. The score of the possibility that two entities are in a certain relationship is defined by the following function:

$$g(e_1, R, e_2) = u_R^T f \left(e_1^T W_R^{[1:k]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R \right) \quad (7)$$

Where f is a standard nonlinear function, $W_R^{[1:k]} \in R^{d \times d \times k}$ Is a tensor, $e_1^T W_R^{[1:k]} e_2$ Is the bilinear tensor product, The result is $h \in R^k$, Each h_i represents a tensor cut $h_i = e_1^T W_R^{[1:k]} e_2$, Other parameters related to R refer to the standard neural network: $V_R \in R^{k \times 2d}$, $U \in R^k$, $b_R \in R^k$, $g(e_1, R, e_2)$ the higher the function score, It shows that the relationship between e_1 and e_2 the entity and R is higher.

The NTN model fully retains the entity semantics implicit in the long tail entity by expressing the long tail entity as the average value of the word vector. In the absence of text description information, the NTN model can still be used to disassemble the entity name. Get entity description information.

Although the NTN model can dig out implicit relationships and entity semantics, it also needs to link the discovered new entities outside the knowledge base to the knowledge base. In this regard, Chen[22] et al. improved the NTN model based on the NTN neural tensor network. The improved model discovers new entities outside the knowledge base by predicting additional real relationships between entities and links them to the knowledge In the library, realize the expansion and completion of the knowledge base. The model uses word vectors in the unsupervised text corpus to initialize the entity representation. At the same time, the model uses a method similar to the long-tail entity representation in the NTN model, using subunits separating entity names with spaces to capture grammatical and semantic information, thereby detecting whether entities outside the database are in a specific relationship. The improvement of the NTN model does not require manual design rules nor additional analysis of other text resources. However, the NTN model is based on the tensor decomposition method. Because this method does not consider the characteristics of the multi-path structure in the knowledge graph, the NTN model can only be used for single-step reasoning and

can only predict the relationship between directly connected entities , And can't mine the deep-level relationship information passed between entities through paths.

ProjE: with the development of knowledge graphs, the amount of information stored in the knowledge base continues to increase, the model feature space of knowledge reasoning is becoming increasingly complex, and the parameter scale is also growing. In order to reduce the parameters, Shi [23] et al. proposed a projection embedding model (Embedding Projection, ProjE) with better effect and smaller parameter scale by simplifying the underlying model architecture. The main method is to treat entity prediction as a multi-candidate ranking problem. Among them, the candidate with the highest ranking score is the entity prediction result, as shown in Figure 2.

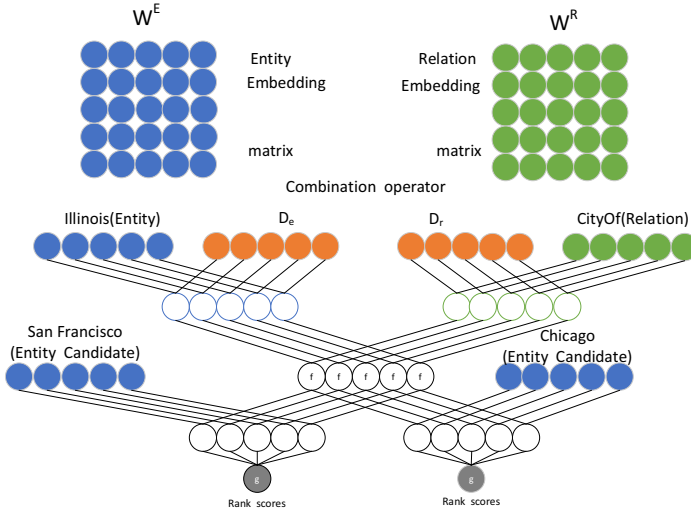


Figure 2. ProjE model

In order to generate the ranking list, this method projects each candidate object onto a target vector generated by the input vector through a predefined combination operator, and calculates the projection similarity. The combination operator is defined as follows:

$$e \oplus r = D_e e + D_r r + b_c \tag{8}$$

The sum is the diagonal matrix, which is the weight of the entity and the relationship respectively, and is the combined deviation. The model defines the embedded projection function as:

$$h(e, r) = g(W^c f(e \oplus r) + b_p) \tag{9}$$

Among them, f and g use sigmoid and tanh as activation functions, respectively, which are the matrix of candidate entities and the projection deviation. Represents the ranking score vector, where each element represents the similarity between a certain candidate entity in the combined input embedding.

From the perspective of the number of parameters, compared with other transfer matrix methods, the number of parameters in the ProjE model is greatly reduced in the process of using combined operations. This model can not only perform entity prediction tasks, but also perform relationship prediction tasks by changing the input. However, because the ProjE model processes the entity and relationship embedding vectors through the combination operator to generate the output vector, the quality of

the word vector preprocessing has a great influence on the combination operation. Not only that, the ProjE model does not use semantic information such as entity descriptions and relationship paths, and the interpretability of the reasoning results is still lacking.

All in all, single-step reasoning based on neural networks attempts to use the powerful learning capabilities of neural networks to model knowledge graph fact tuples, and obtain good reasoning and generalization capabilities. However, the inherent interpretability problems of neural networks still exist. How to properly explain in the application of knowledge graph

The reasoning ability of neural networks is a big difficulty. At present, there are relatively few researches on single-step reasoning based on neural networks, but the high expressive ability of neural networks and its application in other fields, including image processing, text processing, and knowledge. The outstanding performance and high performance of graph structure data fields such as social networks with similar graph structures make the research prospects in this direction broad. How to extend more neural network-based methods in other fields to the field of knowledge graphs has become a subject to be studied in depth in the future. Question. General graph structure data, such as the representation and reasoning of social networks, are nodes, while the representation and reasoning of knowledge graphs focus on nodes (entities) and edges (relationships). Therefore, from general graph structure data based on neural networks, the transfer of methods to the knowledge graph will be a relatively simple breakthrough. At the same time, the research on the interpretability of neural networks needs to be further developed.

3.3 Confluent reasoning

ILP: In the process of rule discovery assisted by distributed representation, the discovery of reasoning rules in the earlier reasoning method of combining rules and distributed is realized by calculating the distributed similarity between relations, and the relationship is represented as the feature vector of corresponding instance. However, the earlier methods do not consider the specific context information, and use an independent method to model the relationship, which ignores the dependency between relationships. In view of the above two problems, Han[24] et al. proposed a context sensitive reasoning rule discovery method. To adopt this method, you need to first build a relationship graph (as shown in Figure 5), and use abstract relationship tuples (such as A buy B) and instantiated features (such as A=Facebook and B=WhatsApp) as the nodes in the relationship graph. The edge in represents the co-occurrence of the abstract relationship tuple and the instantiation feature (such as the edge between A buy B and A=Facebook) or the semantic dependency between the abstract relationship tuple (Such as the connection between A buy B and A purchase B). Then, learn the relationship representation of a specific context based on the relationship graph (as shown in Figure 3, the relationship acquire on the specific context (Google, YouTube) and (children, skill) will be different), and the abstract relationship tuple is related to the corresponding feature context sensitively. The sex score is visualized as a splicing vector, and the correlation score is calculated by restarting random walks from the relation tuple to the feature. Finally, the similarity between the relation vectors is calculated, and the relation pair and the corresponding context larger than the threshold are used as inference rules.

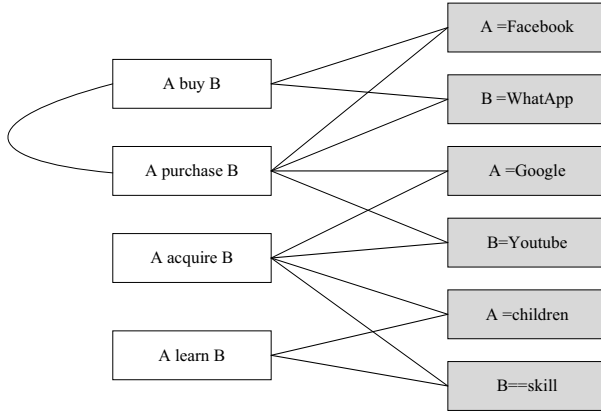


Figure 3 Example of relationship diagram

In the aspect of rule assisted reasoning based on distributed representation, Wang [25] et al. seamlessly embedded logical rules and physical rules in the representation model, formalizing the reasoning method into an integer linear programming problem (ILP). Physical rules include the restriction of entity type and the number of entities. In general logic rules, if another relationship R_2 can be deduced through relationship R_1 , then the two entities connected by relationship R_1 are also connected by relationship R_2 , and the model and rules are integrated by ILP. The objective function is based on the score function of the triple, and any of the above can be used to represent the score function of the model, and each rule is formalized as a condition to be satisfied.

Although this method can improve the reasoning accuracy to a certain extent, the rules in these methods need to be instantiated at a high cost.

DKRL: There are two hybrid models in the hybrid reasoning method of neural network and distributed representation: the first is to use the neural network method to model the external information such as the related external text and entity description, and use the representation model to model the triples in the knowledge map; the first method is to use the neural network method to model the knowledge map and get the output Value is applied to the representation model. Xie[26] et al. thought that the description information of an entity can help to improve the accuracy of entity representation, so a description embedded knowledge representation learning (DKRL) model was proposed, which used the entity description in the knowledge map to predict the entity and relationship. The model uses two kinds of encoders, including continuous bag of words (CBOW) and deep convolution neural network. By learning the description content, this method can not only obtain the structure information in the triple, but also the keywords in the entity description and the text information hidden in the word order. The energy function e of DKRL model is defined as:

$$E = E_S + E_D \tag{10}$$

Where, is the energy function based on structure representation, which is the same as the energy function of model TransE:

$$E_S = \| h_s + r - t_s \| \tag{11}$$

E_D is an energy function based on description representation, which is defined as:

$$E_D = E_{DD} + E_{DS} + E_{SD} \tag{12}$$

E_{DD} is defined as:

$$E_{DD} = \| h_d + r - t_d \tag{13}$$

The head and tail entities are all based on description. EDS and ESD are defined as:

$$E_{DS} = \| h_d + r - t_s \| \tag{14}$$

$$E_{SD} = \| h_s + r - t_d \| \tag{15}$$

One of the head and tail entities is a description based representation, and the other is a structure based representation. The training goal of DKRL model is to minimize the marginal based score function L:

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'} [\gamma + d(h + r, t) - d(h' + r', t')] + \tag{16}$$

Among them, s is the set of positive case triples, which is the set of negative cases. The negative example means that one of H, R and T in the triple (h, R, t) is replaced by another entity or relationship with the wrong triple. γ is the marginal super parameter, which represents the minimum distance margin between positive and negative examples, and D () represents the calculation distance.

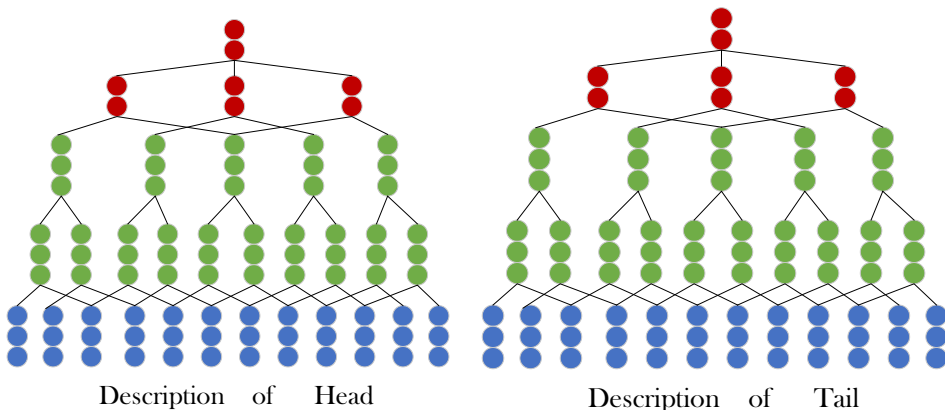


Figure. 4 DKRL model deep convolution neural network framework

The advantage of DKRL model is that it combines the text description information of entity in the process of entity representation, but the disadvantage is that the word vector needs to be preprocessed. The model is sensitive to noise data and needs to be eliminated manually. Moreover, the model is only compared with the TransE model, so it is not referential.

Generally speaking, the hybrid reasoning in the single-step reasoning method is to use a variety of different single-step reasoning methods to achieve complementary advantages. At present, the hybrid single-step reasoning method still stays in the shallow mixing of one method as the main method and the other as the auxiliary method. The hybrid reasoning method is different from other specific methods. It is more like a strategy. It is mainly based on the thorough analysis of various methods to find and mix the methods with complementary advantages. The development lies in the selection of several methods for integration and the mode of integration. Among them, one method is used as the main method and the other as the auxiliary strategy in the hybrid method of rule and distributed representation. There are two main ideas in the hybrid one-step reasoning of neural network and distributed representation: one is to use the existing representation model to learn the triples of knowledge map, and the other is to directly use neural network to model knowledge map and integrate the

learning model. In view of the outstanding performance of neural network in various aspects, it will become a hot spot in future research.

4 Multi step reasoning

The multi-step reasoning method is based on the direct relationship modeling by single-step reasoning, which is a deeper indirect relationship modeling, namely multi-step relationship. Multi step relation is a kind of transitive constraint. For example, there is relationship R_1 between entity A and entity B and relationship R_2 between entity B and entity C. The direct relationship corresponding to these two-step paths is the relationship R_3 between entity A and entity C. Through the introduction of multi-step relationship, more information can be modeled, so the effect of multi-step reasoning method is generally better than that of single-step reasoning method. According to different reasoning methods, multi-step reasoning can also be divided into distributed representation based reasoning, neural network-based reasoning and hybrid reasoning.

4.1 Reasoning for distributed representation

The multi-step distributed representation reasoning method also infers through the vectorized knowledge map. Different from the single-step reasoning method, multi-step relation constraints are introduced in the process of learning vector representation. The introduction of multi-step relationship can make the learned vector representation more helpful to the inference and prediction of entities and relationships. For the case of multiple paths between entities, the contribution difference of paths is not distinguished in the TransE model, and these paths are treated equally no matter how important the practical significance is. In this way, some paths with little practical significance are considered more, which increases the complexity of the model. In view of the above situation, the academia has made a series of improvements.

PTransE: For the drawbacks of unified treatment of paths in the TransE model, Lin et al. [27] proposed the PTransE model, which deals with different paths between entities. PTransE model adds relationship path constraint on the basis of TransE, and models the path through the combination operation of relationship. As shown in Figure 5, the paths pass through the relationships BornInCity, CityInState and StateInCountry in turn. Through the combination operation modeling of relations, the combination operation can be in the form of addition and multiplication of relations on the path, and then the combined path is regarded as the transition between the head entity Steve Jobs and the tail entity UnitedState. For multiple paths between entities, the weight of relational paths is weighed by path constrained resource allocation algorithm [14], which is used to weight the path modeling results.

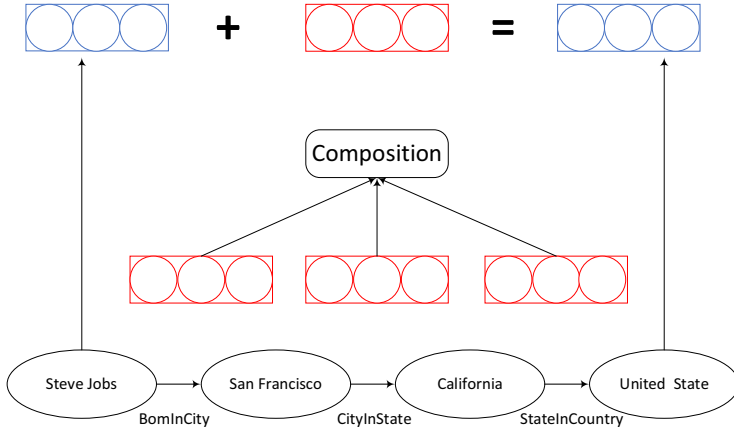


Figure 5. PTransE Path modeling

Specifically, for a relation path $p=(r1, \dots,rl)p=(r1,\dots,rl)$ We define a semantic composition operation and obtain the relational path representation $p=r1, \dots,rl$.

For a multi-step path triple (h, P, t) , we can define its energy function simply by referring to the TransE .

$$(h, p, t) = ||p - (t - h)|| = ||p - r|| \tag{17}$$

When the relation path P and relation R are consistent, the energy function has a lower score, and does not need to consider the vector information of the entity:

$$L(S) = \sum_{(h,r,t) \in S} [L(h, r, t) + \frac{1}{Z} \sum_{p \in P(h,t)} R(p|h,t)L(p,r)] \tag{18}$$

L is the energy function, Z is the normalization factor, and R is the reliability for the entity (h, t) relation path P.

Although PTransE model increases the relationship path constraints and reduces the parameters of combination operation, PTransE model can't deal with some complex relationship forms, such as reasoning from Queen to female, so it can be used as first-order logic to code this reasoning.

GAKE: At present, most of the knowledge representation methods deal with triples separately and PTransE model only introduces multi hop relation information. In this regard, Feng et al. [28] proposed a graph aware knowledge embedding (GAKE), which uses the structural information in the knowledge map to learn the vector representation of entities and relationships.

The GAKE model introduces three kinds of graph context information: neighbor context, path context and edge context. It reflects knowledge attributes from different angles. Meanwhile, it designs the attention mechanism, that is, weight learning of entities and relationships to learn entities or relationships with representative capabilities. Among them, the neighbor context reflects the triple relationship, the neighbor context of an entity is the relationship and tail entity pair in all triples with the entity as the head entity, the neighbor context of the relationship is the head entity and tail entity pair in all triples corresponding to the relationship, and the path context is the entity and relationship on the multi-step path. The edge context of an entity is all the relationships connected with the entity, and the edge context of a relationship is all the entities connected by the relationship. The objective function of each context is the sum

of probability functions of entities or relationships in a given context. Finally, the GAKE model maximizes the weighted sum of the above three kinds of objective functions.

In general, multi-step reasoning based on distributed representation is mainly based on the single-step reasoning based on distributed representation to add multi-step relationship modeling, using supplementary modeling or joint modeling. The method of supplementary modeling is Single-step relationship is the main, multi-step relationship assists learning, used to adjust the empty space

In general, multi-step reasoning based on distributed representation is mainly based on the single-step reasoning based on distributed representation to add multi-step relationship modeling, using supplementary modeling or joint modeling. The method of supplementary modeling is as follows: The single-step relationship is the main, and the multi-step relationship auxiliary learning is used to adjust the position of the vector in the space. The method of co-modeling treats the direct relationship and the indirect relationship equally to obtain a better vector representation, but may introduce cascading errors, that is, the errors caused by the relationship modeling of the intermediate steps on the path will be transmitted Accumulate to the final result. For the common modeling approach, some of the high-quality paths and the corresponding direct relationships actually constitute transitive rules. In this case, multi-step reasoning based on distributed representation can be used to a certain extent It can be seen that while modeling the triples in the knowledge graph, the transitive rules are additionally modeled in the form of vector operations that are convenient for calculation, and the reasoning performance is improved. This requires determining whether the multi-step path and the corresponding direct relationship have a strong correlation. However, PTransE only generally calculates the reliability of multi-step paths in the entire knowledge graph. In fact, a certain path may be an unreliable path for the predicted relationship r_1 , but it is a decisive path for the predicted relationship r_2 . Therefore, considering the reliability of the path and the corresponding direct relationship, more likely and effective transitive rules will be obtained, and the modeling on this basis will further improve the effect. This will also become a small research point.

4.2 Reasoning based on Neural Network

RNN+PRA classifier-b: The fact triples in the knowledge map are not isolated from each other. Many triples have connection relations. The tail entity of one triplet may be the head entity of another. Such two triples can form a five tuple path, and there may be some relationship between the head and tail entities of the five tuples. For example, if there are triples (a, father, b) and (B, father, c), you know that there should be a triple (a, grandfather, c). In order to predict the relationship between the first and last entities of a multi group path, Arvind et al. [29] proposed a multi hop relationship reasoning method, which uses recurrent neural network (RNN) to combine the implied semantics of relationships in paths of any length. In the combination step of each path, the embedding vector of the next hop relation and the vector representing the multi hop path from the beginning to the present entity are combined into an output vector as the input of the next step. After using the multi hop path, RNN outputs a relation vector between the first entity and the last entity of the path, which contains all the information of the relationships in the whole path.

In the combination step of each path, this method uses the embedded vector of the next hop relationship and the vector representing the multi-hop path from the beginning

to the current entity to form an output vector (representing the extended path so far) , As input for the next step. After using the multi-hop path, RNN outputs a relationship vector between the first entity and the last entity of the path, which contains the information of all the relationships on the entire path. As shown in Figure 6, after combining all the relationship vectors along the tuple path "Microsoft (IsBasedIn) Seattle (StateLocatedIn) Washington (CountryLocatedIn) USA" in turn, the above method sets the head entity "Microsoft" and the tail entity "USA" Produce a vector very close to the "CountryOfHeadquarters" relationship.

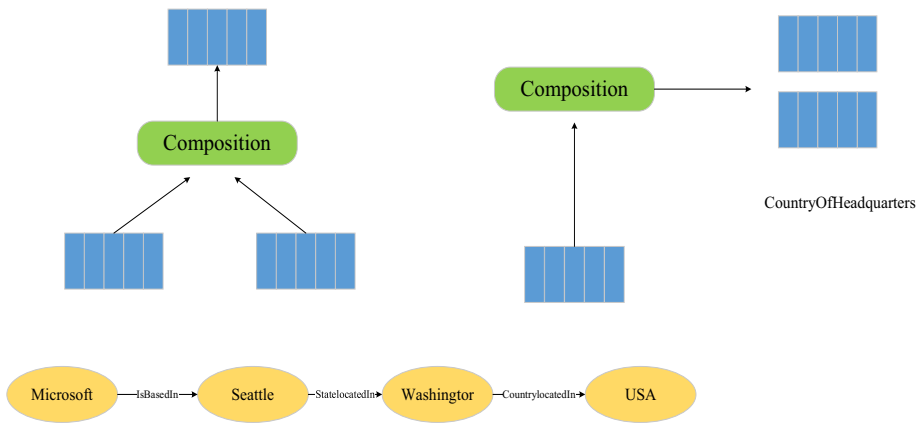


Figure 6 Multiple jump relation inference method

In other words, this method uses all the relationship information in the multi group, combines them in turn and generates the final merging relationship between the head and tail entities. However, considering that a large number of multiple groups in the knowledge map do not necessarily have sequential semantic association, it is difficult to explain the combination reasoning only by using the relationship information on the path of multiple groups, and it is easy to be misled by the semantic differences of different types of relationships on the path in the calculation, resulting in meaningless or unexplained reasoning results.

Path-RNN: In other words, there may be multiple paths between the first and the last entities of a multi-component group formed by multiple triples in the knowledge map. In other words, there may be multiple paths between the first and last entities. The previous multi hop relation reasoning method only uses the single path information of the multivariate group, that is, the probability score is calculated by the combination of the relationship vectors on the path, and other paths are not considered, and the entity on the path is not modeled, but only the "merge" relationship is deduced. In order to improve the above method, Das et al. [30] combined rich symbolic logic reasoning with neural network with strong generalization ability to train a single high-capacity *RNN*. This model alleviates the common problems of Path Ranking Algorithm (*PRA*) that cannot share parameters and a large amount of parameters. It allows the sharing of parameters across multiple relationship types for reasoning. It uses the intermediate paths that exist between entity pairs. Information about entities, and express them as annotated functions to alleviate the issue of entity transparency. Top-k, average and LogSumExp methods are used to add weight to the similarity score of each path. In this

way, the information contained in each path is considered, not just the one with the highest score. However, this method is not suitable for dealing with long text features.

The Path-RNN model is a high-capacity RNN model that allows inference chains across multiple relationship types. It utilizes the information of intermediate entities existing in the path between entity pairs, and alleviates the problem of invisible entities by representing entities as functions of their annotation types. In addition, the author of the article proposes that gathering evidence across multiple paths can improve training speed and accuracy, and shows significant performance improvements through multitasking.

In summary, the multi-step reasoning process based on neural network is a process of directly modeling multi-step path or reasoning. Compared with the single-step reasoning based on neural network, the research work is richer, the interpretability is stronger, and the effect is better. Good. However, the interpretability needs to be further enhanced. Among them, the way of directly modeling the multi-step path, because the multi-step path can be regarded as a relationship or a sequence of relations and entities, it is mainly modeled by RNN. In view of this direction, the research space is relatively small and has related work, and its continued development space is relatively small. The method of directly modeling the reasoning process can simulate the learning and reasoning process of humans, and people have strong learning and reasoning capabilities, so this will be a very popular The direction of research. Moreover, the process of human learning and reasoning is relatively complex, making it a very challenging problem. Based on existing models, further research on how to better model the reasoning process will become a need for future efforts Problems solved.

4.3 Confluent reasoning

In multi-step reasoning, hybrid reasoning is realized by mixing different multi-step methods. Neural network is usually used to mix with other methods because of its strong learning ability and generalization ability. The hybrid reasoning in multi-step reasoning mainly includes path sorting and embedding method, and rule and neural network hybrid reasoning.

RL reinforcement learning method: In order to deal with the problem that it is impossible to find a clear answer in complex query problems; the intelligent algorithm must be able to infer the existing resources and learn to deduce the location of the answer. Specifically, in a large-scale knowledge map, when faced with multi hop reasoning relationship, it is necessary to learn the formula of display reasoning.

Aiming at the above problems, Xiong[31] and others proposed a new controllable multi hop reasoning method, which constructed the path learning process as reinforcement learning. Compared with PRA model, this method uses translation based knowledge-based embedding method to encode RL agent into continuous space, and RL agent conducts reasoning in vector space of knowledge map. The agent extends its path by sampling a relationship, thus taking incremental steps. In order to better guide RL agent learning relationship path, the method also uses strategy gradient training and a new reward function, which jointly encourages accuracy, diversity and efficiency.

The specific task of relational reasoning is to find reliable prediction paths between entity pairs [29]. In this method, the path finding problem is described as a sequential decision problem that can be solved by RL agent. RL system consists of two parts. The

first part is the external environment, which specifies the dynamic interaction between agent and knowledge map. The environment is modeled as a Markov decision process.

The second part of the system is the RL agent, which is represented as the policy network [32], which maps the state vector into the random policy. The parameters of neural network are updated by stochastic gradient descent. Compared with DQN, the RL method based on policy is more suitable for the scenario of knowledge map. One reason is the path finding process of the knowledge map, and the behavior space may be very large because of the complexity of the relationship diagram. This may lead to poor convergence of DQN. In addition, the strategy network can learn the gradient strategy, prevent the agent from falling into some intermediate state, and avoid the problems of value-based methods such as DQN in learning strategy gradient.

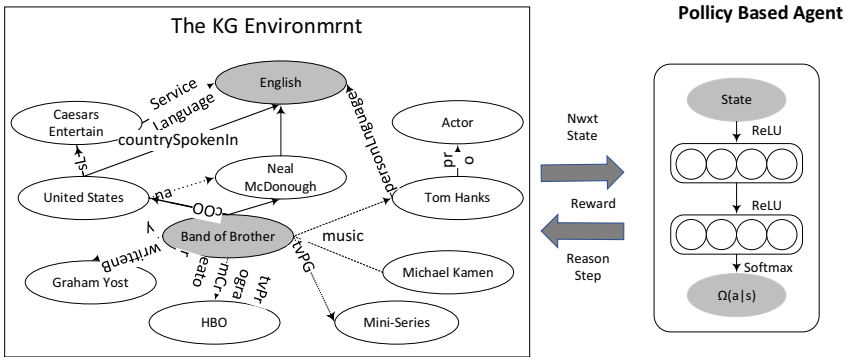


Figure 7 RL system diagram

For the setting of the environment, the number of operations that an agent can perform can be very large, because there are many more wrong sequential decisions than the correct macro volume. To solve the problem of large number of operations, the first reward function is added in RL model.

$$r_{GLOBAL} = \begin{cases} +1, & \text{if the path reaches } e_{target} \\ -1, & \text{otherwise} \end{cases} \quad (19)$$

The purpose of reward function is that when agent R reaches the target after performing a series of behaviors, an offline positive reward + 1 will be given.

In relational reasoning tasks, the short path contains more reliable reasoning evidence than the long path. Short relation chain can also improve the efficiency of reasoning by limiting the length of interaction between RL and environment. Efficient reward is defined as:

$$r_{EFFICENCY} = \frac{1}{length(p)} \quad (20)$$

The path P is defined as a sequence of relations.

The RL agent is trained to find the path of positive samples for each relationship. These training samples (resources, etargets) have similar state representation in vector space, and RL agents tend to find paths with similar syntax and semantics. These paths usually contain redundant information because some of them may be relevant. In order to encourage agents to find diverse paths, we use the cosine similarity between the current path and the existing path to define a diversity reward function.

$$r_{DIVERSITY} = -\frac{1}{|F|} \sum_{i=1}^{|F|} \cos(p, p_i) \quad (21)$$

Where $p = \sum_{i=1}^n r_i$ is the embedding path $r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_n$ of the relation chain. Unlike previous random walk based pathfinding models, RL model allows us to control the attributes of the paths we find. In many path based reasoning methods, these effective paths can also be used as a substitute for PRA.

This model uses a reinforcement learning framework to improve the performance of relational reasoning in KGs. Specifically the author trains an RL agent to find a reasoning path in the knowledge base. Unlike previous pathfinding models based on random walking, the RL model allows us to control the properties of the path found. In many path-based reasoning methods, these effective paths can also be used as a substitute for PRA. For the two standard reasoning tasks, the RL path is used as the reasoning formula, which is usually better than other methods.

In future research, the model can introduce adversarial learning to provide the possibility of better rewards with the human-defined reward function used in the model. Instead of designing rewards based on path characteristics, a discriminant model can be trained to give rewards. In addition, in order to solve when KG does not have enough reasoning paths, we can apply the RL framework to joint reasoning based on KG triples and text mentions.

Neural LP: The reasoning of hybrid rule and neural network mainly transforms the rule into vector operation, which is applied to the neural network method with strong learning ability to realize a differentiable model. Yang et al. [32] proposed a fully differentiable system. For the first time, we learned an end-to-end differentiable model Neural LP (Neural logic programming) by combining the parameters and structures of first-order logic rules. In NeuralLP, each entity is associated with a fixed one hot vector, and each relationship is associated with a fixed $\{0,1\}$ operation matrix. The logic rule reasoning is formalized as matrix multiplication. The score function of the entity is weighted and correlated with the confidence degree of the product of the relational operation matrices on all corresponding paths. Neural LP designs a neural controller system with attention mechanism and storage. The learning process combines the differentiable operations used in TensorLog in sequence. The parameters to be learned are rule set and its corresponding confidence degree. Since each confidence level is associated with a specific rule, enumerating rules is a discrete task, and direct learning is difficult to formalize into a differentiable process.

In order to solve this problem, the multiplication and addition operations of the score function are exchanged, that is, add first and then multiply. Firstly, the confidence weighted sum of operation matrices is carried out from the perspective of relationship, and then multiplied. However, the resulting paths are all the same length, related to the number of multiplications. Therefore, the auxiliary storage vector, storage attention vector and operation attention vector are introduced, and the learning is carried out in the form of circulation. Among them, the auxiliary storage vector is initialized to the given entity, followed by the intermediate reasoning results obtained by each step; the storage attention vector stores the weight of each step until the current step; the operation attention vector stores the weight of each TensorLog operation. Finally, the model calculates a weighted average of all storage vectors, and the weight is the storage attention vector. In this way, use attention to select the appropriate rule length. The whole process is implemented by LSTM. In each step of reasoning, the attention mechanism is used to select the subset of TensorLog operation, and then the

content is selected from the storage. The result of operation is the storage vector of the step and added to the storage.

The model treats the KGC task as a sorting problem, and puts the candidate entities on a combined embedding vector representing the known part of the input triples, and arranges the ranking score vector in descending order. In order to improve the prediction performance, the author uses a list The ProjE mutation of the formula optimizes the collective ranking score vector. In the training process, only the direct connection and the length of 1 path are used, which has a relatively simple 2-layer structure, but is better than complex models with richer parameters or feature sets. Compared with other models, the current work of this model does not require any pre-trained embeddings and has much fewer parameters than related models. Finally, the author proves that ProjE performs better than existing methods in fact checking tasks. In the future development, the model can use the embedded projection model proposed in this paper to adapt to more complex neural network models, such as RNN and CNN. But this will inevitably add additional complexity. How to reduce complexity is the focus of future research.

In general, hybrid multi-step reasoning is richer in content than hybrid single-step reasoning. In addition, after recent years of development, distributed reasoning methods have more effective hybrid modes. However, the current hybrid reasoning is still limited to the mixture of the two methods. Among various hybrid methods, the reasoning of hybrid rules and neural networks has a lot of room for development, the high accuracy and interpretability of the rule method and the strong strength of the neural network method Learning and high generalization capabilities enable the combination of the two to obtain a high-accuracy differentiable model, avoiding the computational problems of traditional rule methods, and to a certain extent, increasing the interpretability of neural network methods. In the future, mixed rules Reasoning with neural networks will be worthy of more in-depth research, which can be further studied from the perspective of hybrid modes and the specific manifestations of rules and neural networks.

5 Comparison of experiments and methods

5.1 Data set and evaluation index

In order to compare the quality of the model, this article selects the FB15K-237 dataset and the WN18 dataset. The FB15K-237 dataset is a subset of Freebase and contains 237 relationships and 14K entities. The WN18 dataset contains 18 relationships and 40K species of entities. The results of fact prediction and link prediction are used as evaluation indicators to compare some knowledge reasoning methods. The subtasks of link prediction include entity prediction and relationship prediction, the purpose of which is to predict the missing part of the triple.

Entity forecast: Entity prediction is to predict the missing entities in triples. The specific test methods are as follows:

- 1) Extract the triad $\langle h, R, t \rangle$ from the test set;
- 2) The head entity and tail entity in the dataset are replaced respectively, and the triplet $\langle e_n, R, t \rangle$, n is the number of entities in the dataset;

3) The newly replaced triples are scored and sorted from high to low reliability, and the lower the score, the better;

4) Repeat the first three steps until all triples in the test set are tested.

There are two evaluation indexes: Meanrank and evaluation index hits@10. Meanrank is the average rank of all test triples in the test dataset; hits@10 is the probability of the top 10 triples in the test dataset. A good link prediction method should have a lower mean rank and a higher link prediction method hits@10.

In addition, it should be noted that if a triple knowledge map constructed during the test exists, that is, the triple is actually correct, it is reasonable to rank it before the test triples $\langle h, R, t \rangle$. In order to eliminate the influence of this problem, the correct triples existing in training set, verification set and test set are eliminated before ranking each test triplet. It is no doubt that the evaluation result of "filter" is more important than "raw".

Relationship prediction: The method of relation prediction is similar to entity prediction, except that the target of replacement is changed from entity to relation. Due to the small number of relationships in the knowledge map, hits@10, it is difficult to compare the advantages and disadvantages of different methods hits@1 Instead hits@10.

5.2 Comparative analysis of experimental results

When training the model, the learning rate α is set to $\{0.01, 0.1, 1\}$, the spacing l is set to $\{0.25, 0.5, 1\}$, the vector dimension d is set to $\{20, 50, 100\}$, the balance rate between the models is set to η It is $\{0.01, 0.1, 1, 10\}$, and the nearest neighbor structure order n is set to $\{1, 2, 3\}$. After many experiments, the optimal parameter configuration is as follows: On the FB15K data set, $\alpha = 0.01, l = 1, d = 50, \eta = 0.1, n = 2$; on the WN18 data set, $\alpha = 0.01, l = 2, d = 20, \eta = 0.1, n = 2$. The experimental results are as follows:

Table 1: entity prediction results on FB15k-237

Method	MeanRank		Hits@10(%)	
	Raw	Filter	Raw	Filter
TransE	243	125	34.9	47.1
TransD	194	91	53.4	77.3
TransR	198	77	48.2	68.7
PTransE	200	54	51.8	83.4
ProjE	124	34	54.7	88.4
DKRL	200	113	44.3	57.6

Table 2: entity prediction results on WN18.

Method	MeanRank		Hits@10(%)	
	Raw	Filter	Raw	Filter
TransE	263	251	75.4	89.2
TransD	224	212	79.6	92.2
TransR	238	225	79.8	92.0
PtransE	242	92	50.6	82.2
ProjE	152	56	79.7	92.3
DKRL	212	125	54.3	87.6

From the experimental data in the table, whether it is in the FB15K data set or the WN18 data set, each model has its own outstanding side and certain defects. Among them, the ProjE model performs on both Raw and filter. It's the best. In the Trans series model, the PTransE model performs better, slightly worse on Raw, and best on filter. The reason for this result is that some correct triples interfere during sorting, but in the filter process. It is to deal with this interference.

When there are not enough paths between entities, RL model and PRA model may give bad results. For example, for the FilmWrittenBy relationship, the RL model only finds four unique inference paths, which means that there is not enough reasoning evidence in the knowledge map. Another observation is that RL achieves better performance on Nell datasets. By analyzing the paths found from the map, we find that Nell dataset has more short paths than FB15k-237, and some of them are just synonyms of reasoning relations.

Table 3:Number of reasoning paths used by PRA and RL model

Tasks	#of Reasoning Paths	
	PRA	RL
worksFor	247	25
teampplaySports	113	27
TeamPlaysInLeague	69	21
...		
Average	137.2	20.3

Table 3 shows some comparisons of the number of inference paths. From the data in the table, it can be found that compared with PRA model, RL agents with pre-defined reward functions can select better paths, and filter out similar or unrelated paths, so as to enhance the reasoning strength.

Next, we provide quantitative results supporting our claim that modeling the entities along a KB path can improve reasoning performance. The last section of Table 4 lists the performance gain obtained by injecting information about entities. We achieve the best performance when we represent entities as a function of their annotated types in Freebase (Single-Model + Types) ($p < 0.005$).

Table 4: Model performance comparison

model	Performance (%MAP)
ILP	51.13
PRA	64.43
Path-RNN	68.43
RNN+PRA classifier-b	61.17
RL	65.41
Neural LP	73.28

Table 4 compares the performance of the six model methods. When training all paths, NeuralLP performed significantly ($p < 0.005$) better than other models. The accuracy of NeuralLP model is as high as 73.28%, the accuracy of RL reinforcement learning model is 65.41%, and the accuracy of Path-RNN model is 68.43%.

From the comparative analysis of the above experimental results, it can be found that the mixed model is better than the single model in terms of accuracy and model complexity.

6 Summary

Generally speaking, the single-step reasoning method is based on the fact tuples in the knowledge map, while the multi-step reasoning method adds the multi-step path constraints on the basis of the single-step reasoning modeling method. The expression ability of the multi-step reasoning method is generally stronger than that of the single-step reasoning method, and the inference prediction effect is relatively better. The spatial distribution characteristics of knowledge map are fully considered in the modeling of single-step reasoning method based on distributed representation. The spatial distribution of knowledge map is modeled by fine-grained method, which has stronger expression and reasoning ability. The research on this kind of methods is still relatively small, and it needs to further explore the spatial distribution characteristics of knowledge map, explore more modeling methods, and expand to multi-step reasoning. Among the subclasses of single-step reasoning and multi-step reasoning, neural network-based reasoning and hybrid reasoning are relatively good methods. With the continuous development of neural network, more in-depth research still needs to be carried out. Further enhancing the interpretability of neural network for the task of reasoning is also a major difficulty. Hybrid reasoning attempts to use the advantages of various reasoning methods to obtain better reasoning performance. Generally, the effect of ensemble learning is better than that of single model. However, the current hybrid reasoning is limited to the mixing of the two methods. How to mix multiple complementary methods to further improve the reasoning ability needs further research.

References

- [1] Wang YZ, Jia YT, Liu DW, Jin XL, Cheng XQ. Open Web knowledge aided information search and data mining. *Journal of Computer Research and Development*, 2015, 52(2): 456-474.
- [2] Jin XL, Wah BW, Cheng XQ, Wang YZ. Significance and challenges of big data research. *Big Data Research*, 2015, 2(2): 59-64.
- [3] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A nucleus for a Web of open data. In: *Proc. of the 6th Int'l Semantic Web Conf. Berlin, Heidelberg: Springer-Verlag, 2007. 722-735.*
- [4] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In: *Proc. of the 2008 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2008. 1247-1250.*
- [5] Ding Jianhui, Jia Weijia. Overview of Knowledge Graph Completion Algorithms[J]. *Information and Communication Technology*, 2018 (1): 57-62.
- [6] Xu ZL, Sheng YP, He LR, Wang YF. Review on knowledge graph techniques. *Journal of University of Electronic Science and Technology of China*, 2016, 45(4): 589-606.
- [7] Gardner, Matt, and Tom Mitchell. "Efficient and expressive knowledge base completion using subgraph feature extraction." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015.
- [8] Clark, Herbert. *Linguistic Processes in Deductive Reasoning*. *Psychological Review*, 1969, 76(4): 387.
- [9] Arthur W Brian. *Inductive Reasoning and Bounded Rationality*. *The American Economic Review*, 1994, 84(2): 406-411.
- [10] van Hoek, Remko, et al. "Abductive reasoning in logistics research." *International journal of physical distribution & logistics management* (2005).
- [11] Gentner Dedre. *Structure-mapping: A theoretical Framework for Analogy*. *Cognitive Science*, 1983, 7(2): 15-170.
- [12] Guan Saiping, Jin Xiaolong, Jia Yantao, et al. Research progress of knowledge reasoning oriented to knowledge graph[J]. *Journal of Software*. 2018, 29(10): 74-102.
- [13] Pearl Judea, Azaria Paz. *Graphoids: A Graph-Based Logic for Reasoning about Relevance Relation* ECAI, 1985: 357-363.

- [14] Rudolf Kadlec, Ondrej Bajgar, Jann Kleindienst Knowledge Base Completion: Baselines Strike Back. *ACL 2017*, 69-74.
- [15] Zhang Youmin, Liu Li, Fu Shun, et al. Entity Alignment Across Knowledge Graphs Based on Representative Relations Selection. *ICSAI 2018*, 2018:1056-1061.
- [16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. (2013).
- [17] Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In: *Proc. of the Advances in Neural Information Processing Systems*. Red Hook: Curran Associates Inc., 2013. 2787-2795.
- [18] Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: *Proc. of the 29th AAAI Conf. on Artificial Intelligence*. Menlo Park: AAAI, 2015. 2181-2187.
- [19] Ji G, He S, Xu L, Liu K, Zhao J. Knowledge graph embedding via dynamic mapping matrix. In: *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics*, Vol. 1. Stroudsburg: ACL, 2015. 687-696.
- [20] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, et al. Convolutional 2D Knowledge Graph Embeddings. *AAAI*, 2018:1811-1818.
- [21] Socher R, Chen D, Manning CD, Ng A. Reasoning with neural tensor networks for knowledge base completion. In: *Proc. of the Advances in Neural Information Processing Systems*. Red Hook: Curran Associates Inc., 2013. 926-34.
- [22] Chen Danqi, Socher R, Manning C D, et al. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *J. Computer Science*, 2013. 11(1):392-399.
- [23] Shi, Baoxu, and Tim Wenginger. "ProjE: Embedding projection for knowledge graph completion." *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.
- [24] Han X, Le S. Context-Sensitive inference rule discovery: A graph-based method. In: *Proc. of the 26th Int'l Conf. on Computational Linguistics*. Stroudsburg: ACL, 2016. 2902-2911.
- [25] Wang Q, Wang B, Guo L. Knowledge base completion using embeddings and rules. In: *Proc. of the 24th Int'l Joint Conf. on Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers, 2015. 1859-1866.
- [26] Xie R, Liu Z, Jia J, Luan H, Sun M. Representation learning of knowledge graphs with entity descriptions. In: *Proc. of the 30th AAAI Conf. on Artificial Intelligence*. Menlo Park: AAAI, 2016. 2659-2665.
- [27] Lin Y, Liu Z, Luan H, Sun M, Rao S, Liu S. Modeling relation paths for representation learning of knowledge bases. *arXiv Preprint arXiv: 150600379*, 2015
- [28] Feng J, Huang M, Yang Y, Zhu X. GAKE: Graph aware knowledge embedding. In: *Proc. of the 26th Int'l Conf. on Computational Linguistics*. Stroudsburg: ACL, 2016. 641-651.
- [29] Neelakantan A, Roth B, McCallum A. Compositional vector space models for knowledge base completion. *J. Computer Science*. 2015:1-16.
- [30] Das R, Neelakantan A, Belanger D, McCallum A. Chains of reasoning over entities, relations, and text using recurrent neural networks. *arXiv Preprint arXiv: 160701426*, 2016.
- [31] Xiong W, Hoang T, Wang W Y, et al. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. *C. empirical methods in natural language processing*, 2017:564-573.
- [32] Yang F, Yang Z, Cohen WW. Differentiable learning of logical rules for knowledge base reasoning. *arXiv Preprint arXiv:170208367*, 2017.