

# A Knowledge Graph Data Expansion Method Based on Relational Propensity Categories with Legal Applications

Xiaoqin Liu<sup>1</sup>, Chenou Xu<sup>2</sup>, Fei Pu<sup>2</sup>, Berlin Yang<sup>2</sup>, Shi Meng<sup>2,\*</sup>

<sup>1</sup>Zhejiang Gongshang University Hangzhou College of Commerce, Hangzhou 311599, China;

<sup>2</sup>School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

**Abstract.** By introducing the definition of propensity categories of relations, the implicit information in the knowledge graph is mined and the MCBE (Maximum Clique Based Expansion) algorithm is used for data expansion. Experimental results show that with baseline models TransE, RotatE, HAKE and Complex on FB15K dataset, its MRR and Hits@1 metrics are improved by (7.9%, 9.6%), (4.2%, 3.3%), (2.7%, 4.8%) and (1.7%, 2.4%), respectively. Experiments are also conducted on the FB15K, YAGO3-10, NELL-995 and DBpedia50 datasets using the TransE model as a baseline, and its MRR and Hits@1 metrics are improved on the above datasets by (7.9%, 9.6%), (0.3%, 27.7%), (20.1%, 100%), (4%, 38.7%), respectively. Finally, the MCBE algorithm is applied to the self-constructed knowledge graph of "anti-drug law" and its MRR and Hits@1 metrics are improved by (10.6%, 12.3%). The experimental results show that MCBE algorithm improves the prediction accuracy of legal knowledge graph.

## 1 Introduction

With the development of big data and the Internet, huge amounts of data are constantly being generated in all areas of life and industry. The first step in finding a suitable application for this data is to choose a reasonable way of storing it. Knowledge bases, also known as knowledge graphs, are born here, and are stored in the form of the Resource Description Framework (RDF) published by the World Wide Web Consortium, which defines a knowledge base to be stored as a common standard form of a tuple [1]. A tuple usually contains two entities and one relation, expressed as (head entity, relation, tail entity), but also as (subject, relation, object). Like the way nodes and edges are connected in graph theory, knowledge graphs in this way form a semantic network capable of revealing relationships between entities. Search engines such as Google and Bing are using knowledge graphs to enhance their search experience, and social networking applications such as Facebook are using it to mine connections between users. Knowledge graph research also includes, but is not limited to, semantic parsing, entity alignment, information extraction, relationship extraction, entity linking, and

various other tasks [2-4]. In addition to these, Knowledge Graphs have been widely used in practical applications in a variety of fields such as intelligent search, recommender systems, deep Q&A, natural language processing, and so on.

There are already many open source knowledge graph databases available in the Internet, e.g., DBpedia, YAGO, FreeBase, and so on [5-7].

In terms of the judicial field, knowledge mapping also has a good application scenario. The court system has very rich case data, but it lacks effective organization, and the key knowledge is difficult to be extracted, analysed and utilized. Through the construction of legal knowledge graph, it is possible to split a case into a large number of event triples based on causality, and to link many cases together. By constructing and utilizing the legal knowledge graph, the accuracy of legal applications such as crime prediction, legal intelligent quiz, and legal knowledge query can be improved.

Knowledge graphs, while capable of preserving a large number of triples, also bring new challenges to learning and using them. Knowledge graphs have a huge amount of data and there are three main challenges that need to be addressed:

\*Corresponding author's email: [mengshi@zjgsu.edu.cn](mailto:mengshi@zjgsu.edu.cn)

### (1) Computational efficiency issues

The storage form of the knowledge graph is equivalent to that of a graph in a data structure, where each entity needs to be represented by a node, and each relationship between entities has to be represented by an edge, which results in a knowledge base that usually has tens of thousands of nodes, as well as hundreds of thousands of edges between nodes. Learning from the knowledge graph comes down to the complexity and time cost of the algorithms. Conventional algorithms are often not capable of learning large-scale knowledge graphs, and can only be used to learn small and medium-sized knowledge graphs.

### (2) Data sparsity problem

Data in knowledge graphs also tends to suffer from sparsity due to its possible long-tail effect: a small number of nodes have a large number of relationships, and most nodes have only one or two pieces of ternary information. For example, the DBpedia50 dataset has 49,900 entities and 32,388 triples, each entity involves only 0.65 triples on average, and most of the entity objects have only one triple information that can be utilised for learning. For these entities, their information can only be inferred through the information of their related nodes or the related paths in the knowledge graph, which makes learning extremely difficult, and the prediction and complementation tasks involving them tend to have extremely low accuracy rates.

### (3) Scalability issues

The reality is that the knowledge base must be constantly updated and improved, but many knowledge graph algorithms do not take into account the knowledge graph has a subsequent complementary situation exists, in the learning and training is completed and put into use, often can not be added to the new entities and relationships, or re-learning of the cost is huge. This also puts forward new requirements for the scalability of knowledge graph related algorithms.

The current mainstream research approach to knowledge mapping through KRL (Knowledge Represent Learning), i.e., representing triples with low-dimensional embedding vectors, has been able to basically control the computational efficiency problem in the first challenge in a reasonable amount of time. [8]

This paper focuses on the data sparsity problem proposed above, inspired by methods such as AMIE (Association Rule Mining Under Incomplete Evidence), which introduces external information to enhance knowledge graph learning. [9] Most of the existing knowledge graph learning models only consider the direct

information in the knowledge graph and ignore the large amount of implicit information in the knowledge graph. By introducing the concepts of "relationship category" and "category tendency of relationship", and combining them with traditional graph algorithms, a knowledge graph data expansion embedding algorithm based on maximal clusters is proposed.

The algorithm expands the amount of data in the knowledge graph dataset by mining the implicit information in the knowledge graph itself, thereby expanding the amount of data in the knowledge graph dataset without introducing external information, and allowing the model to learn more fully in the subsequent learning of the knowledge representation, thus improving the performance and accuracy of the model so that it can be better applied to downstream tasks.

The algorithms in this paper have been experimented and validated on a self-constructed knowledge graph of legal document cases related to the Drug Law of the People's Republic of China, in addition to a general-purpose dataset, to obtain an improvement in accuracy on the offence prediction task.

The main contributions of this paper are as follows:

1. The algorithm is dataset-independent, applicable to all knowledge graphs, there is no requirement for the composition of the knowledge graph dataset, and all of them have a certain effect, and there is more considerable accuracy improvement on certain knowledge graphs, which indicates that this algorithm has good scalability and adaptability.

2. This algorithm is data desensitised, and the training learning process does not need to know the actual entity and relationship names, i.e., it can be learnt for graphs with only graph structures.

3. Relative to the learning process, the present algorithm is relatively simple and does not increase the time complexity of the model, adding only a few minutes to ten minutes of computing time for small to large knowledge graph datasets.

4. The algorithm combines the graph algorithm with the knowledge graph for the first time, proving that the traditional algorithm can also play a role in the training of complex models nowadays, and providing certain ideas for the direction of the subsequent combination of the graph algorithm and the knowledge graph.

5. Experiments and analyses were carried out on the constructed legal knowledge graph, which proved that the

algorithm in this paper can practically enhance the application effect of knowledge graph in law.

## 2 Related work

### 2.1 Embedding-based learning model for knowledge graph representation

#### 2.1.1 Semantic Matching Models Based on Tensor Decomposition

Semantic models based on tensor decomposition pass to represent the potential semantics of entities as well as relations in a vector space. The more basic ones of this type are RESCAL bilinear model, RESCAL uses full-rank matrices as the representation of relations, and all that exists between entities and relations are matrix operations, thus allowing deep interaction of information between entities and relations. [10] However, it also brings the following drawbacks, (1) The one using RESCAL model is easy to overfitting and possesses lower generalisation ability. (2) The RESCAL model is stored in matrix form, and its operational complexity is of order of magnitude with the number of entities and relations, making it difficult to be applied to large-scale knowledge graphs.

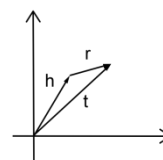
Improving on this problem, the DistMult model RESCAL model is based on relaxing the constraints on the relation matrix. [11] The relationship matrix is simplified into a diagonal matrix, which drastically reduces the complexity of the operation and can fetch better results. However, because of the simplification of the relation matrix, it results in DistMult only being able to portray symmetric relations in the knowledge graph, and not being able to solve other types of relations.

The ComplEx model, on the other hand, introduces for the first time a complex domain for knowledge mapping based on DistMult. [12] It is able to portray both symmetric and asymmetric relationships.

#### 2.1.2 Translation-based modelling

TransE is the first translation-based model, the translation model is inspired from the (King-Man) vector in Word2Vec should be approximately equivalent to the (Queen-Woman) vector, and each triad (Entity 1, Relationship, Entity 2) in the knowledge graph dataset is regarded as a process of translating from the head Entity 1 vector to the tail Entity 2 through the Relationship  $r$ . That is, the Relationships are regarded as translations from Entity 1 to Entity 2, and for each triad (Entity 1,

Relationship, Entity 2), TransE expects Entity 1+Relationship=Entity 2. This is shown in Fig. 1, where  $h$ ,  $r$ , and  $t$  denote the head Entity, Relationships, and Tail Entity, respectively. [13]



**Fig. 1.** Example of TransE model.

TransE model is relatively simple, with fewer parameters and lower computational complexity, but it can make good use of the potentially complex semantic links between entities and relations, and has amazing effects on modelling large-scale sparse graphs, which can be said to have become a representative and important model in knowledge graphs. However, TransE is still insufficient for the portrayal of relations, and is overstretched for the three kinds of complex relations: 1-N, N-1, and N-N. Complex relationships here are categorised according to the number of entities connected to the head and tail of the relationship, e.g. 1-N means that each head entity will correspond to N tail entities in that type of relationship.

TransH, on the other hand, in order to solve the difficulty of the TransE model in dealing with complex relations, proposed the idea of letting an entity have different representations under different relations. [14] Each entity vector is regarded as a projection of different relation planes in specific different triples. It's able to solve the many-to-many relationship problem existing in the knowledge graph to some extent.

TransR, on the other hand, first proposed the concept of an entity being an aggregate of multiple attributes, i.e., an entity can have multiple attributes, but for a particular relation, it may involve only some of the attributes of that entity. [15] TransR believes that each relation has its own semantic space, and for each ternary ancestor in the Knowledge Graph, TransR projects its entity into the corresponding relationship space before training.

TransD constructs a dynamic mapping matrix for each entity-relationship pair by simultaneously considering the diversity of entities and relationships. [16] Compared with TransR/CTransR, TransD significantly reduces the

algorithm complexity due to fewer parameters and no matrix-vector multiplication is required.

TranSparse on the other hand considers that entities and relationships in the knowledge graph are heterogeneous and unbalanced, TranSparse uses sparse matrices to model the relationships to solve the problem of heterogeneity. [17] Also two separate sparse transfer matrices are created for each head entity and tail entity to solve the imbalance problem.

The RotatE rotational model can be seen as a variant of the translation model, which models relationships in complex space as rotational operations between head and tail entities. [18] It has been theoretically proven to model a wide range of situations such as symmetric/antisymmetric relationships (e.g. marriage, filial relationship), flipped relationships (superior/subordinate relationships), and combinatorial relationships (mother's husband = father) in relationships in the knowledge graph.

Whether it is a semantic matching model based on tensor decomposition or a translation-based model, both are designed only from the perspective of semantic space, with the goal of improving the applicability and expressiveness of the model or reducing the complexity of the model. No consideration is given to the fact that knowledge graphs come from real data in reality, ignoring their rich implicit logic and potential semantic connections between entity relationships.

## 2.2 Data-enhanced learning models for knowledge graph representation

AMIE (Association Rule Mining Under Incomplete Evidence), an association rule mining algorithm based on incomplete knowledge base, is a rule-learning-based reasoning algorithm that emphasizes on automated rule-learning methods to quickly reason out rules with high confidence from large-scale knowledge graphs, and then join them to the knowledge graph for reasoning tasks. For the first time, it proposes to add new data to the knowledge graph. The method focuses on how to assess the validity of the rules, generally using a confidence level approach, the higher the confidence level of a rule, indicating that there are multiple instances of the rule in the knowledge graph, it is more likely to be accurate.

On the basis of AMIE, RUGE uses a recurrent training approach, which adds rules into the knowledge graph dataset as soft rules and then trains them in a traditional way. The soft rules are extracted again as a result of the

completed training and trained again. With this reciprocation to a suitable threshold, good results were also achieved.

The above two methods prove that the learning training effect of knowledge graph models can be improved by adding useful information to the knowledge graph. However, there are two drawbacks: (1) is that wrong information may be added during the training process, and if the threshold is not well controlled, too much wrong information added may instead reduce the accuracy of the model. (2) is that these methods require some human intervention.

## 3 Methodology of this paper

### 3.1 Entity categories in the knowledge graph

Knowledge graphs are constructed by natural language processing or manually, and the entities involved are basically from the same direction or domain. For example, in the case of a film knowledge graph, the entities involved may be the name of the film, the director, the actors, the dubbing, and so on. For example, for the names "Inception" and "Creed", they both belong to films, so they may exist in the knowledge graph in the form of a triad of (Inception, director, Nolan) and (Creed, director, Nolan). These two triples can be generalised in the form of (film title, director, Nolan), so their entity category is film. The same entity can have more than one category attribute, e.g. "Xu Zheng" is an actor in the film "I am not the God of Medicine" and a director in the film "Me and My Hometown".

### 3.2 Relational propensity categories in the knowledge graph

The traditional category definition of a relationship is the category definition of the meaning of the word relationship itself, for example, in the triad (Inception, director, Nolan) and (Silver Age, author, Wang Xiaobo), the relationship "director" and the relationship "author" may be relatively similar in the traditional definition and may belong to the same category. The relationship between "director" and "author" may be similar in the traditional definition, and may belong to the same category. Because they are also in the form of a "work" corresponding to a "creator", i.e. they are both in the form of (work, director or author, creator). Therefore, the common category of director and author can be summarized as "creative professions".

In this paper, a new definition of "relationship tendency category" is proposed: i.e., the category of a relationship is not determined by itself, but by the categories of its head and tail entities, in the form of (head tendency, tail tendency). The head tendency represents the tendency of the relationship to the category of the head entity, and the tail tendency represents the tendency of the relationship to the category of the tail entity.

So for the above two triples, it is possible to convert the above two relations from the form (work, director or author, creator) to the form (film name, director, director's name) and (book name, author, author's name) respectively. The preferred category of the relation "director" is (film name, director's name) and the preferred category of the relation "author" is (book name, author's name).

So the relations "director" and "author" are not of the same tendency category, whereas the relations "author" and "translator" are of the same tendency category. The two relations "author" and "translator" have the same tendency category (book name, author's name).

### 3.3 Data expansion methods based on relational propensity categories

Based on the concept of "relationship propensity category" proposed above, this paper proposes a data expansion method based on relationship propensity category, the core of which is to find out whether the propensity category of each relationship is the same through the implicit links contained in the triples in the knowledge graph dataset, if the propensity categories of the two relationships are basically the same, it can be considered that the two relationships involving triples can be replaced with the head entity or the tail entity to achieve the expansion of the dataset. If the tendency categories of the two relationships are basically the same, it can be considered that the ternary groups involved in these two relationships can be replaced by the head entity or the tail entity, so as to achieve the expansion of the dataset. If only the "head tendency" of the two relationships is the same, then the head entities of their related triples can provide mutual expansion, and similarly, if only the "tail tendency" of the two relationships is the same, then the tail entities of their related triples can provide mutual expansion. For example, if the propensities of the two relationships "situated" and "located" are essentially the same, the triad (West Lake, situated, Hangzhou) can be expanded with (West Lake, located, Hangzhou) to give the relationship "located" additional propensities. The relation "located in" can be learnt by

expanding the triad with (West Lake, Located, Hangzhou), thus giving the relation "located in" more related triples.

The method has several advantages:

1. The method can be applied to all knowledge graph datasets, i.e., no matter what kind of knowledge graph dataset can be used, and there is no requirement for the number of triples, entities and relationships in the knowledge graph dataset.

2. The method is data desensitised and can be applied to knowledge graph datasets that are represented only by codes and not by actual names.

3. The method has a low time complexity, and the time complexity of its data expansion method is approximately equal to  $O(n)$ , where  $n$  is the number of triples in the knowledge graph dataset.

The difficulty of the method is that a suitable method needs to be found to find relationships with consistent propensity categories in a knowledge graph and noisy data needs to be avoided.

Since common knowledge graph datasets basically have a number of triples in the hundreds of thousands, and the number of their relationships is only a few hundred to a few thousand, leading to the fact that each relationship may involve hundreds or even thousands of triples, making ordinary methods such as clustering difficult to apply to determine whether the propensity classes of relationships are consistent, and prone to introducing noisy data.

So in this paper, from the graph algorithm, the maximal clusters algorithm of graphs is introduced to find the tendency categories of relations, and a maximal clusters-based embedding algorithm for knowledge graph data expansion is proposed to ensure that the data added to the knowledge graph is reliable.

### 3.4 Explanation and characteristics of the Maximum Clique Algorithm

The Maximum Clique Algorithm also known as Maximum Clique Problem is a classical combinatorial optimization problem in graph theory which is defined as follows:

Given an undirected graph  $G = (V, E)$ , where  $V$  is a non-empty set, called the vertex set, and  $E$  is the set of unordered binary groups consisting of elements in  $V$ , called the edge set, edges in an undirected graph are all unordered pairs of vertices, and a graph  $U$  is said to be a complete subgraph of  $G$  if  $U \subseteq V$  and there is  $(u, v) \in E$  for any two vertices  $u, v \in U$ .

A complete subgraph  $U$  of  $G$  is a clique of  $G$  if and only if  $U$  is not contained in a larger complete subgraph of

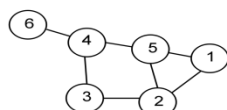
G. The largest clique of G is the one that contains the largest number of vertices in G.

U is said to be a null subgraph of G if  $U \subseteq V$  and  $(u, v) \in E$  for any  $u, v \in U$ . A null subgraph U of G is an independent set of G if and only if U is not contained in a larger null subgraph of G. The maximal independent set of G is the one with the highest number of vertices contained in G.

For any undirected graph  $G = (V, E)$ , its complement  $G' = (V', E')$  is defined as  $V' = V$  and  $(u, v) \in E'$  if and only if  $(u, v) \notin E$ .

If U is a complete subgraph of G, then it is also a null subgraph of  $G'$  and vice versa. Thus, there is a one-to-one correspondence between the groups of G and the independent sets of  $G'$ . Exceptionally, U is a maximal clique of G if and only if U is a maximal independent set of  $G'$ .

The goal of the maximal clusters algorithm is to find a subgraph of the graph in which the points are completely connected, and there is no way to add any point from the graph in such a way that the subgraph still maintains this property. The nodes 1, 2, and 5 in Figure 2 below are a maximal clump:



**Fig. 2.** Example of the maximum clique

The nodes in the same maximal clusters, which all have very similar nature characteristics, are introduced to the entity categories of the knowledge graph, which can be considered as similar categories.

### 3.5 Construction of a Knowledge Graph Based on the Anti-Drug Law of the People's Republic of China

In this paper, based on 500 case documents under the relevant categories of "anti-drug law", a legal knowledge map of "anti-drug law" is constructed by extracting textual information using natural language processing and combining with manual screening.

The main types of entities involved in the knowledge graph are key information such as "name of person", "type

of drug" and "grams". Four types of relationships are defined, namely "drug trafficking", "providing shelter", "drug possession" and "drug delivery".

## 4 Knowledge Graph Data Expansion Algorithm Based on Relational Propensity Categories

With the maximal clusters algorithm and the concept of "relationship propensity category" proposed above, the algorithm in this paper can find the largest clusters of relationships with the same "relationship propensity category" in the knowledge graph dataset, so as to complement each other with the triads of relationships belonging to a cluster, thus improving the learning effect of the final model on each relationship and entity. The algorithm in this paper can find the largest group of relationships with the same "relationship tendency category" in the knowledge graph data set, so as to complement each other for the relationships involved in a group, thus improving the learning effect of the final model on each relationship and entity, and improving the accuracy of the model.

This algorithm firstly analyses the triples in the training set of the knowledge graph dataset to obtain the entity list of each relationship, and then compares and filters it with the threshold S through the similarity calculation method, and calculates the head and tail entity similarity table of each relationship respectively. Through the head entity similarity table, each relationship is regarded as a "node" in the graph, and the relationships with which the head entity is related are regarded as having "edges", so as to obtain the head tendency similarity graph of the relationship. Similarly, do the same for the tail entity similarity table. Then, the maximal clique algorithm is used on the head tendency similarity graph and tail tendency similarity graph respectively, so as to obtain the relationship groups with consistent head tendency or consistent tail tendency. The data involved in the relationships of each maximal clusters can be complemented with each other.

In this paper, the above Maximum Clique Based Data Expansion Algorithm is called MCBE (Maximum Clique Based Expansion Algorithm) method. This chapter explains the detailed steps of this algorithm based on the FB15K dataset, mainly using the head entity as an example.

The overall flow of the algorithm is divided into the following steps:

1. Count the head tendency and tail tendency of each relationship based on the triads in the training set.
2. Calculate the similarity between the head and tail tendencies of the relationships, the more similar two relationships are considered to be related, and calculate the two complete relationship correlation table.
3. Using the head and tail relationship tables, each relationship is treated as a "vertex", and two related relationships are treated as if they had "edges". A new relationship graph is formed to be used in the Maximum Cluster Algorithm to find the clusters in the graph (the clusters found are the communities between the relationships).
4. Relationships in the same group are considered to have the same head or tail tendency. Further, relationships with the same head and tail tendencies are considered equivalent.
5. Completion of the triples in the original data set based on the data involved in the relationships in each of the largest clusters.

The reason for not using the similarity of relationship propensity categories directly as a judgement of similarity is as follows: assuming that in a dataset, there is similarity of propensity categories between relationship A and relationship B, similarity of propensity categories between relationship B and relationship C, and similarity of propensity categories between relationship C and relationship D, using similarity alone as a judgement criterion will result in the expansion process, where relationships A, B, C, and D are equal, while the actual part of similarity between each relationship is different. In the expansion process, relations A, B, C and D are equated, while the actual similarity between each relation is different. For example, "watermelon" and "passion fruit" belong to the category of "fruit", "watermelon" and "chocolate" belong to the category of "fruit", "watermelon" and "chocolate" belong to the category of "fruit", "watermelon" and "chocolate" belong to the category of "fruit". "Chocolate" both belong to the category "sweet food", but "passion fruit" and "chocolate" are not equivalent. Adopting this simple approach will result in the model converging all the relationship embeddings when learning entity and relationship embeddings, which will lead to a very poor model.

The nature of maximal clusters ensures that nodes belonging to the same maximal clusters have almost identical properties, and the maximal clusters algorithm to find the maximal clusters in the relational association graph

ensures that all relations belonging to a cluster have a highly similar tendency to the head or tail of the relation. That is, in other words, judging by similarity can only speculate that the relationships involved have certain connections, while judging by the maximal clusters algorithm can ensure that all relationships belonging to the same clusters are near-synonyms or even synonyms, avoiding the addition of noisy data.

#### 4.1 Calculation of the propensity to relate

Firstly, we read in the data in the training set, in which the data are in the form of "head entity, relationship, tail entity", each row is a ternary, and desensitise the read-in data to keep only the ids, i.e., transform the names of the entities and relationships containing textual information into a purely encoded id table. We also transform the triples in the training set, test set and validation set according to the id table, and finally transform (head entity, relationship, tail entity) into a pure number similar to (381, 8, 789) to save the form. Table 1 below:

**Table 1.** Data set format description

Serial number	Header entity id	Relationship id	Tail entity id
1	381	8	789
2	6203	17	18253
3	837	84	9213
.....	.....	.....	.....

Two dictionary tables are used to record the head entity and tail entity involved in each relationship in the training set. Taking the head entity as an example, data similar to the form of Table 2 below is obtained after processing:

**Table 2.** Statistics on header entities involved in relationships

Relationship id	Number of entities involved	Involved entity id
1200	4	3123, 1034, 58, 5733
725	3	12123, 4388, 11087
.....	.....	.....

Table 2 represents 1200 The head entities involved in this relationship are numbered 3123, 1034, 58, and 5733.

The specific algorithm is as follows Algorithm 1:

Algorithm 1: Statistical algorithm for head/tail entities involved in relationships
Input: training set triad S, one triad $i(h, r, t)$ for each row.
Output: a statistical table of the entities involved in the relationship.
1. FOR $i \in S$ .
2: IF $i.h$ NOT IN $i.r\_head$ .
3: $i.r\_head.ADD(i.h)$
4: IF $i.t$ NOT IN $i.t\_tail$ .
5: $i.r\_tail.add(i.t)$

Similarity calculation to obtain relationship association table

Using the data of relationship head tendency and relationship tail tendency obtained in the previous step, i.e., Table 2, the similarity of the tendency of the head \ tail entity involved in each relationship is computed two by two, and the similarity computation method adopted here is the cosine similarity method, and below the threshold S is regarded as no tendency similarity between these two relationships. The threshold S is usually inversely proportional to the number of relationships R in the dataset, and has been tested experimentally, in most cases the threshold is more appropriate to be taken as 0.85, and if the number of relationships is small (e.g. less than 100), the threshold is more appropriate to be taken as 0.9. Take head entity dumping as an example, the specific algorithm is as follows Algorithm 2:

Algorithm 2: Similarity calculation algorithm
Inputs: header entity list $H_a$ for relation A, header entity list $H_b$ for relation B, threshold $S = 0.85$ .
Output: similar relationship numbered table $S\_Table$ .
1. Similarity = $(H_a \cap H_b) / (H_a \cup H_b)$
2: IF Similarity > S:
3: $S\_Table[A].ADD(B)$

With Algorithm 2, the relational association table for the FB15K dataset can be obtained as shown in Table 3 below:

**Table 3.** Relational association table for the FB15K dataset (based on header entities)

Relationship id	List of relationship ids that are similar to it
1	8
2	6, 7, 32, 38, 59, 97, 102, 103, 141, 145, 158, 162, 173, 230

3	8, 12, 27, 47, 74
6	2, 7, 32, 38, 55, 59, 97, 102, 103, 141, 173, 230
7	2, 6, 32, 38, 55, 59, 97, 102, 103, 141, 145, 158, 162, 173, 230
.....	.....

Each row represents the corresponding similarity of a relation. For example, relationship 3 has a similar head-to-head tendency with relationships 8, 12, 27, 47, and 74, and is considered to be a related relationship.

According to Table 3, each relationship id is regarded as a "vertex", and if there is a similar relationship between two relationships, an "edge" is added to them to form a new relationship graph.

## 4.2 Maximum clique algorithm to find relationship communities

Based on the relational association graph, the maximal clusters algorithm described in section 3.5 of this paper was used, with the specific implementation choosing the Bron-Kerbosh algorithm, which is generally considered to be more efficient than other similar algorithms, to find the maximal clusters between relations in this new relational association graph. The final maximal clusters found are shown in Table 4 below:

**Table 4.** The maximum clique table obtained from the FB15K dataset (based on head entities)

Mission number	Largest group
1	8, 77, 123
2	9, 57, 75, 235
3	12, 28, 147
4	36, 169, 117
.....	.....

Relationships that fall within one of the largest regiments between two and two head tendencies are basically considered to be completely similar.

## 4.3 Data set expansion based on the largest clique table

Based on the maximal clusters table of relations, the training set is expanded, such as (Entity 381, relation 8, entity 789) and (Entity 482, relation 77, entity 67), because 8 and 77 belong to the same cluster in terms of the head



tendency, and the ternary groups involved in them can be complemented with each other to obtain (Entity 482, relation 8, entity 789) as well as (Entity 381, relation 77, entity 67), which increase the the number of involved triples, which can enable the model to better learn the characteristics of these some relations.

The same algorithm is used to process the operations on the tail tendency of the relation to obtain the maximum clique table of the tail tendency to be expanded.

For relations where both the head and tail tendencies are identical, this paper considers them to be fully equivalent relations that can be used to expand the dataset directly by substituting the relations of the involved triples.

## 5 Results

In this chapter, FB15K, YAGO-3, NELL-995, and DBpedia50, which are the widely used standard datasets for knowledge graphs, are selected as the experimental data sources, and four representative mainstream embedded learning models for knowledge graphs, TransE, RotatE, HAKE, and Complex, are used in conjunction with the MCBE expansion method proposed in this paper to Conducting experiments. This chapter mainly adopts the metrics enhancement in the link prediction experiment results to verify the effectiveness of this algorithm.

Finally, this chapter demonstrates the effectiveness of this paper's algorithm in the field of legal knowledge graph by conducting experiments and analyses on the constructed knowledge graph of "anti-drug law".

This chapter mainly covers data set, benchmark model selection, link prediction, evaluation metrics, experimental setup and result analysis, and "anti-drug law" knowledge mapping experiment and analysis.

### 5.1 Introduction to the dataset

The datasets used for the algorithms in this chapter are the FB15K dataset, the YAGO3-10 dataset, the NELL-995 dataset and the DBpedia50 dataset. They are chosen for the following reasons: (1) These are mainstream datasets commonly used in the evaluation of various knowledge graph algorithms. (2) These datasets all contain a large number of entities and relations, which are difficult to learn, and all of them contain a variety of relation classes including symmetric, asymmetric, and inverse relations, which cover a variety of types of representative datasets.

where FB15K was introduced in the 2013 paper TransE by Bordes et al. It is a subset of Freebase, which

contains about 14,951 entities with 1,345 different relationships. Freebase is a large open-source knowledge graph dataset in which knowledge is filled in by users.

YAGO3, on the other hand, is a huge semantic knowledge base derived from Wikipedia, WordNet, and GeoNames. Currently, YAGO3 holds knowledge about more than 10 million entities (e.g., individuals, organisations, cities, etc.) and contains more than 120 million facts about these entities. The huge amount of data makes learning from the YAGO3 dataset very difficult to portray with simple models. YAGO3-10 is a subset of the YAGO3 dataset.

NELL-995 comes from the results of 995 iterations of NELL (Never-Ending Language Learning system), a semantic machine learning system developed by a team of researchers at Carnegie Mellon University.

The DBpedia Knowledge Base, on the other hand, contains a variety of descriptions of 5.9 million things, including everything from geography, demographics, music, movies, games, companies, organisations, species, diseases, etc. DBpedia50 is a subset of the DBpedia Knowledge Base.

The details of each dataset are given in Table 5 below, containing the number of entities and the number of relationships involved in the dataset, as well as the number of triples included in the training, validation, and test sets, respectively:

**Table 5.** Data set statistics

Data set	Number of entities	Relations	Training set	Validation set	Test set
FB15K	14951	1345	483142	50000	59071
YAGO3-10	123000	37	1,000,000	5000	5000
NELL-995	75492	200	149678	543	3992
DBpedia50	49900	654	32388	399	10969

### 5.2 Benchmark model selection

In this chapter, three representative pairs of knowledge graph embedded representation learning models, TransE, RotatE and Complex, are used to test the results of the data augmentation algorithm. The three benchmark models are described below:

### 5.2.1 Translation-based model TransE

Given a training set  $S$ ,  $S$  consists of the triad  $(h, r, t)$ ,  $(h, t)$  belongs to the entity set  $E$ , and  $r$  belongs to the relation set  $R$ . The size of the embedding dimension is set according to the parameters of the model  $k$ . This model is characterised by the requirement that  $h + r \approx t$ , learns the embeddings by minimising the marginal ranking function on the training set with L1 or L2 regularisation. Its loss function is given in equation (1) below:

$$L = \sum_{(h,r,t)} \sum_{(h',r',t')} [\gamma + d(h+r,t) - d(h'+r',t')] \quad (1)$$

Where  $[x]$  represents the positive part, which is a marginal parameter greater than zero. While  $h'$  and  $t'$  are negative sample triples generated by randomly replacing the head entity or tail entity by the positive sample triples, the generated negative sample triples are required to be absent from the original dataset, and  $d(h+r, t)$  is the similarity measure function.

TransE is one of the more basic models, only able to represent 1-to-1 relationships predominantly, but its design is simpler and can be used as a baseline model for evaluating comparisons.

### 5.2.2 The rotation-based model RotatE

RotatE views relations as rotation vectors between entities. By making the difference between the head entity and the relationship after performing the rotation operation and the tail entity as a function of its distance score, i.e., the following equation (2):

$$d_r(h, t) = \|h \circ r - t\| \quad (2)$$

Negative samples that are not part of the entities present in that knowledge graph are obtained through self-adversarial negative sample sampling. Self-adversarial negative sample sampling refers to the inclusion of a probability to determine the generation probability of negative samples by means of a probability generation formula  $P$ , instead of uniform sampling. It is implemented by taking a positive sample triad and probabilistically replacing its head or tail with other entities to generate a new triad (i.e., negative sample), and this negative sample cannot be the same as the existing positive sample in the

knowledge graph. The probability generation formula is as follows (3):

$$p(h'_j, r, t'_j | \{(h_i, r_i, t_i)\}) = \frac{\exp \alpha f_r(h'_j, t'_j)}{\sum_i \exp \alpha f_r(h_i, t_i)} \quad (3)$$

$P$  is the calculated sampling probability,  $\alpha$  is the set sampling rate, and  $h_i, r_i, t_i$  is the head entity, relationship and tail entity of the  $i$ th positive triad respectively.  $h'_j, r, t'_j$  is the generated  $j$ th negative triad.

By using a probability generating formula, the sampling of negative samples can be made more balanced and efficient, which is more suitable for model training than random sampling methods.

The negative sampling loss function is optimised so that the difference between the means of the distance score functions of the positive samples minus the distance score functions of the negative samples is as large as possible. Hierarchical category information is also added to make the relationship embeddings with the same hierarchical category as similar as possible.

The loss function is given below in equation (4):

$$L = -\log \sigma(\gamma - d_r(h, t)) - \sum_n^{n-1} p(h'_j, r, t'_j) \log \sigma(d_r(h'_j, t'_j) - \gamma) \quad (4)$$

where  $\gamma$  is the boundary value and  $\sigma$  is the Sigmoid activation method.

The tensor decomposition based model Complex

The ComplEx scoring function is defined as equation (5):

$$f_r(h, t) = \text{Re}(h^T \text{diag}(M_r) \bar{t}) \quad (5)$$

where  $h$  and  $t$  are both represented by complex numbers,  $\bar{t}$  denotes the conjugate complex of  $t$ , and  $\text{Re}(\bullet)$  denotes the real part of the obtained complex. ComplEx is able to solve both symmetric and asymmetric relations simultaneously. ComplEx is the first model that introduces a complex number approach in knowledge graph embedding.

### 5.3 link prediction

Link prediction is a common downstream task in knowledge graph representation learning, which can be

subdivided into link prediction for head entities (?) and link prediction for tail entities (head entity, relationship, tail entity) based on the location of the predicted entities, relationship, tail entity) and link prediction to tail entity (head entity, relationship, tail entity). For example (Hangzhou, province, ?) The correct answer of the tail entity in (Hangzhou, Province, ?) is "Zhejiang", then the model needs to compute the possible tail entity vectors from the head entity vector of "Hangzhou" and the relationship vector of "Province" and match the closest tail entity as The prediction is considered correct if the closest tail entity is the "Zhejiang" tail entity vector.

## 5.4 Evaluation indicators

The commonly used evaluation metrics in link prediction tasks are MR, MRR, Hits@1, Hit@3, and Hit@10.

MR (Mean Rank) refers to the average ranking of the correct entity in link prediction, and the smaller MR value represents the better performance of the model link prediction, which is calculated as follows (6):

$$MR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} rank(q_i) \quad (6)$$

MRR (Mean Reciprocal Rank) refers to the inverse of the average rank of the correct entity in the link prediction, the larger the value of MRR, the better the prediction performance of the model, which is calculated as follows (7):

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank(q_i)} \quad (7)$$

Hit@1, Hit@3, and Hit@10 refer to the proportion of the model's top 1, 3, and 10 ranked candidate targets on the link prediction task that contain the correct answer.

### 5.4.1 Experimental setup

Datasets: The algorithm uses FB15K, YAGO3-10, NELL-995 and DBpedia50, which are widely used datasets in the field of knowledge graph, for the experiments.

Benchmark Models: In order to demonstrate the effectiveness of this algorithm on a variety of models, the benchmark models are selected as TransE, a translation-based model, RotatE and HAKE, a rotation-based model, and Complex, a tensor decomposition-based model. These models relate to the three dominant approaches in

knowledge graph embedding, including the classical models (TransE, Complex) and recently proposed models (RotatE, HAKE).

Evaluation metrics: The evaluation metrics are MR, MRR, Hit@1, Hit@3, Hit@10, which are commonly used in the knowledge graph link prediction task, and are obtained by testing the trained model on the test set for the link prediction task. MRR refers to the mean reciprocal rank, and the interval is (0, 1), the bigger represents the better the model effect. MR refers to the mean rank of mean reciprocal rank, which is the opposite of MRR, and the smaller this item is, the better the model is. Hit@1, Hit@3, Hit@10 refers to the proportion of the top 1, 3, and 10 ranked candidate targets of the model that contain the correct answer on the link prediction task.

Hyperparameter setting: this algorithm uses hyperparameter search to determine the best results for each benchmark model. Where the same benchmark model uses the same training method, loss function and regularisation method, optimiser as well as initialisation method and maximum number of iterations on both the original and expanded datasets. The following hyperparameters are mainly considered:

- 1) Embedding dimension  $d$  (128, 200, 256, 400, 512, 1024)
- 2) Batch size (256, 512, 1024)
- 3) Negative sample size  $n$  (128, 256)
- 4) Adaptive sampling rate (0.5, 1.0)
- 5) Learning rate  $lr$  (0.00005, 0.0001, 0.0005, 0.001)
- 6) Number of iterations: 80,000, 120,000, 150,000

All the experimental results in this chapter were obtained by running in the same machine environment, and some of the results may have some deviation from what has been reported in other papers, which may be mainly due to the limited RAM and video memory of the machine, which prevented the testing of the experimental results at higher embedding dimensions, larger batch sizes, and larger negative sampling sizes.

## 5.5 Analysis of experimental results

In this section, experiments based on multiple benchmark models on the FB15K dataset and experiments under different dataset expansions under the TransE model are conducted, and through the comparison of the results, it is found that the experimental results of the present algorithm are better than the original results in all aspects. It is proved that the model proposed by this algorithm can be widely applied on top of various datasets and various benchmark

models, and is a general dataset-independent and model-independent knowledge graph data expansion embedding algorithm.

### 5.5.1 Multi-model experiments based on the FB15K dataset

Under the FB15K dataset, TransE, RotatE, HAKE, and Complex, which are different modelling approaches, have achieved better results than the original dataset under the expanded dataset of this algorithm. As shown in Table 6 below, where those with Clique-beginning are the results of the models under the dataset expanded by this algorithm:

**Table 6.** Results of expansion algorithm for each model on FB15K dataset

Model name	MRR	MR	Hits@1	Hits@3	Hits@10
TransE	0.673	48	0.582	0.735	0.827
Clique-TransE	<b>0.726</b>	<b>40</b>	<b>0.638</b>	<b>0.789</b>	<b>0.862</b>
RotatE	0.763	47	0.721	0.815	0.876
Clique-RotatE	<b>0.795</b>	<b>45</b>	<b>0.745</b>	<b>0.827</b>	<b>0.880</b>
HAKE	0.661	44	0.548	0.749	0.838
Clique-HAKE	<b>0.679</b>	<b>43</b>	<b>0.574</b>	<b>0.759</b>	<b>0.844</b>
Complex	0.769	43.4	0.701	0.815	0.885
Clique-Complex	<b>0.782</b>	<b>43.1</b>	<b>0.718</b>	<b>0.822</b>	<b>0.886</b>

From the experimental results in the above table, it can be seen that the present algorithm is model-independent, and using four different models on the same dataset, all of them can get some enhancement effect, and the enhancement of the two indexes, MRR and Hits@1, is more obvious.

### 5.5.2 Experiments on expansion of different datasets under TransE model

On the other hand, using the TransE model as a benchmark, this algorithm was experimented on the FB15K, YAGO3-10, NELL-995, and DBpedia50 datasets, respectively. It is proved that the algorithm can achieve certain results on a variety of datasets, as shown in Table 7 below:

**Table 7.** Results of expansion algorithm for TransE model on different datasets

Data set	Modelling	MR	MR	Hits@1	Hits@3
FB15K	TransE	0.673	48	0.582	0.735
	Clique-TransE	<b>0.726</b>	<b>40</b>	<b>0.638</b>	<b>0.789</b>
YAGO3-10	TransE	0.447	2337.9	0.357	0.494
	Clique-TransE	<b>0.448</b>	<b>2243</b>	<b>0.456</b>	<b>0.498</b>
NELL-995	TransE	0.279	7162	0.133	0.401
	Clique-TransE	<b>0.335</b>	<b>10398</b>	<b>0.266</b>	<b>0.419</b>
DBpedia50	TransE	0.100	15299	0.031	0.147
	Clique-TransE	<b>0.104</b>	<b>15374</b>	<b>0.043</b>	<b>0.148</b>

From the above table, it can be seen that the model of the present algorithm has some enhancement effect on several datasets with different characteristics. Especially, the Hit@1 enhancement on FB15K, YAGO3-10 and NELL-995 is huge.

The FB15K dataset contains 1345 kinds of relations, and the relations exist to satisfy symmetric, antisymmetric, inverse relations and other situations, more suitable for the expansion of this algorithm, we can see that the enhancement of the indicators are more significant. YAGO3-10 dataset has fewer relations, there are only 37 kinds of relations, and the ternary group involving each relation is more, which needs to raise the threshold of the algorithm for screening, and the final relations obtained are relatively fewer. Relatively fewer. NELL-995 has 200 kinds of relations, and the training set has 149678 triples, which has a relatively reasonable ratio, and each relation has a large number of related triples that can be analysed, and according to this paper's algorithm, we can find out a lot of large maximal clusters, so this algorithm's hit @1 result is improved by about double. While DBpedia50 as a more complex dataset is more difficult to learn, this algorithm has only a small improvement, indicating that the expansion of this algorithm on the dataset with less response to its own characteristics is less obvious, but there is still a certain degree of improvement.

The above can generally indicate that the present algorithm is more general and can be used on most of the datasets with some enhancement results.

### 5.6 Experiments based on the "drug law" knowledge map

Offence prediction is a practical downstream application of the link prediction task on the legal knowledge graph, which refers to predicting the offence for which a defendant will be sentenced based on the description of the case and the factual part of the legal document.

The benchmark model used in this subsection is the TransE model, while still using having MR, MRR, Hits@1, Hit@3, Hit@10 as the evaluation metrics for the experimental results, and all other experimental parameter settings are the same as before, where the ones with the beginning of Clique- are the results of the model with the expanded dataset of this algorithm:

**Table 8.** Results of the TransE model for the "Drug Law"

Model name	MRR	MR	Hits@1	Hits@3	Hits@10
TransE	0.329	83	0.307	0.351	0.434
Clique-TransE	<b>0.364</b>	68	<b>0.345</b>	<b>0.378</b>	<b>0.463</b>

As can be seen from the results in the above table, the experimental results of TransE model on the knowledge graph of "anti-drug law" after the expansion of this paper, there is a certain degree of improvement in MR, MRR, Hits@1, Hit@3 and Hit@10. The improvement in MRR and Hits@10 is especially obvious. After analysing the expanded knowledge map of "anti-drug law", the improvement of Hits@10 is obvious because through the expanded algorithm in this paper, similar cases are supplemented with each other, for example, the relationship between "grams of drugs and the judgement result", which can be used as a reference for many judgement documents. For example, in the relationship between "grams of drugs and the judgement result", there are many judgement documents for reference, which makes a more reasonable prediction order, resulting in a significant increase in Hits@10.

The results of the model can be used to improve the results of the "Intelligent Legal Quiz" task in addition to the "Judgement Result" task.

## 6 Experimental setup

The system and environment configuration of the machine used for the experiment is attached at the end of this chapter for reference:

System: Windows 10 Professional

Processor: Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz

Graphics: NVIDIA GeForce RTX 2080 Ti

Memory: 16GB

python version: Anaconda-based python 3.7

Pytorch version: pytorch 10.2

## 7 Conclusion

This paper firstly describes the research motivation of the algorithm, focuses on the data sparsity problem in the current knowledge graph learning, combines the idea of introducing external information, puts forward the definitions of "relationship tendency" and "relationship tendency category" and proposes an embedding algorithm based on "relationship tendency category" and provides a detailed algorithm description. The definition of "relationship propensity" and "relationship propensity category" is proposed, and based on "relationship propensity category", a knowledge graph data expansion embedding algorithm based on maximal clusters is proposed and described in detail. The experimental part introduces each dataset and evaluation index, and the experimental design and result analysis are carried out on different benchmark models of the same dataset and different datasets of the same benchmark model, which proves the validity and universality of the algorithm, and finally, the experiments on legal knowledge graph prove the validity of the algorithm of this paper in the field of law.

This algorithm is more innovative in the following ways:

1. The algorithm is dataset-independent, applicable to all knowledge graphs, has no requirements on the composition of knowledge graph datasets, and has good scalability and adaptability.
2. This algorithm is data desensitised, and the training learning process does not need to know the actual entity and relationship naming, i.e., it can be learnt for graphs with only graph structure.
3. The present algorithm is relatively simple and does not increase the time complexity of the model.

4. The algorithm combines the graph algorithm with the knowledge graph for the first time, which provides certain ideas for the direction of the subsequent combination of graph algorithm and knowledge graph.

5. Experiments and analyses were carried out on the constructed legal knowledge graph, which proved that the algorithm in this paper can practically enhance the application effect of knowledge graph in law.

The points raised in this paper still have much room for research. For example, for the analysis of the overall characteristics of the knowledge graph how to better combine with the local information, and the knowledge graph of the same model should be how to do to improve the characteristics of different knowledge graphs, rather than designing a model to try to apply multiple knowledge graphs. At the same time, it also puts forward a new thinking space for the application of graph algorithms on knowledge graphs, and how other graph algorithms can be combined and improved on knowledge graphs still need to be thought about.

Questions that could be pursued in this paper in the future are.

1. Improve the data expansion methodology to accommodate more knowledge graph datasets, which is currently more generally effective for knowledge graphs with a small number of relationships.

2. It is possible to consider how other graph algorithms can be combined with knowledge graph learning algorithms.

3. How to better integrate the content of laws and judgement instruments and apply the algorithms of the present paper to broader areas of law than those related to "drug law".

## Acknowledgment

This work was partly supported with research grants from the National Key R&D Programme of China (Grant No:2018YFB1403202) and the National Natural Science Foundation of China (Grant No:62172366).

## References

1. Miller E. An introduction to the resource description framework[J]. *Bulletin of the American Society for Information Science and Technology*, 1998, 25(1): 15-19.
2. Sun X X, Construction of Subject Knowledge Graph Based on Latent Semantic Analysis[D]. Central China Normal University, 2013.

3. Su J L, Wang Y Z, Jin X L, et al. Entity Alignment of Knowledge Graph Combining Semantic and Structural Information[J]. *Journal of Shanxi University: Natural Science Edition*, 2019, 42(01):23-30.
4. Zhao Hai-Xia, Li Lei, Wu Xin-Dong, et al. Information extraction for knowledge graph[J]. *Data Mining*, 2020, 10(4):21.
5. Auer S, Bizer C, Kobilarov G, et al. Dbpedia: A nucleus for a web of open data[M]//*The semantic web*. Springer, Berlin, Heidelberg, 2007: 722-735.
6. Fabian M S, Gjergji K, Gerhard W. Yago: A core of semantic knowledge unifying wordnet and wikipedia[C]//16th International World Wide Web Conference, WWW. 2007: 697-706.
7. Xu K, Reddy S, Feng Y, et al. Question answering on freebase via relation extraction and textual evidence[J]. *arXiv preprint arXiv:1603.00957*, 2016.
8. Ribes D, Bowker G C. Between meaning and machine: Learning to represent the knowledge of communities[J]. *Information and Organization*, 2009, 19(4): 199-217.
9. Ding B, Wang Q, Wang B, et al. Improving knowledge graph embedding using simple constraints[J]. *arXiv preprint arXiv:1805.02408*, 2018.
10. Nickel M, Tresp V, Kriegel H P. A three-way model for collective learning on multi-relational data[C]//*Icml*. 2011.
11. Yang B, Yih W, He X, et al. Embedding entities and relations for learning and inference in knowledge bases[J]. *arXiv preprint arXiv:1412.6575*, 2014.
12. Trouillon T P, Bouchard G M. Complex embeddings for simple link prediction: U.S. Patent Application 15/156,849[P]. 2017-11-23.
13. Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. *Advances in neural information processing systems*, 2013, 26.
14. Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]//*Proceedings of the AAAI Conference on Artificial Intelligence*. 2014, 28(1).
15. Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]//*Twenty-ninth AAAI conference on artificial intelligence*. 2015.
16. Xiong S, Huang W, Duan P. Knowledge graph embedding via relation paths and dynamic mapping matrix[C]//*International Conference on Conceptual Modeling*. Springer, Cham, 2018: 106-118.
17. Ji G, Liu K, He S, et al. Knowledge graph completion with adaptive sparse transfer matrix[C]//*Thirtieth AAAI conference on artificial intelligence*. 2016.
18. Sun Z, Deng Z H, Nie J Y, et al. Rotate: Knowledge graph embedding by relational rotation in complex space[J]. *arXiv preprint arXiv:1902.10197*, 2019.