IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Knowledge Graph Completion: A Review

ZHE CHEN, YUEHAN WANG, BIN ZHAO, JING CHENG, XIN ZHAO, AND ZONGTAO DUAN
School of Informaiton Engineering, Chang'an University, Xi'an, CO 710064 China

Corresponding author: Zongtao Duan (e-mail: ztduana@chd.edu.cn).

This work was supported in part by the Funds for Key Research and Development Plan Project of the Shaanxi Province, China, under Grant 2017GY-072, 2019ZDLGY17-08, 2020ZDLGY09-02.

ABSTRACT Knowledge graph completion (KGC) is a hot topic in knowledge graph construction and related applications, which aims to complete the structure of knowledge graph by predicting the missing entities or relationships in knowledge graph and mining unknown facts. Starting from the definition and types of KGC, existing technologies for KGC are analyzed in categories. From the evolving point of view, the KGC technologies could be divided into traditional and representation learning based methods. The former mainly includes rule-based reasoning method, probability graph model, such as Markov logic network, and graph computation based method. The latter further includes translation model based, semantic matching model based, representation learning based and other neural network model based methods. In this paper, different KGC technologies are introduced, including their advantages, disadvantages and applicable fields. Finally the main challenges and problems faced by the KGC are discussed, as well as the potential research directions.

INDEX TERMS Knowledge graph; knowledge graph completion; entity prediction; relation prediction; deep learning

I. INTRODUCTION

The concept of knowledge graph was first proposed by Google in 2012, which is defined as a large-scale knowledge base composed of a large number of entities and relationships between them. In recent years, knowledge graph, as a semantic network, has been widely used in natural language processing, intelligent question answering system, intelligent recommendation system and so forth. Together with big data and deep learning, knowledge graph now has become one of the core driving power for the development of artificial intelligence [1].

Although commonly used large-scale knowledge graphs such as Freebase [2], DBpedia [3] and YAGO [4] contain millions of entities and relationships, there are still many missing facts, and many implicit relationships between entities have not been fully excavated, resulting in incomplete structure and content of knowledge graphs [5]. Therefore, the research on knowledge graph completion is proposed to complete the knowledge graph and expand its scale by predicting the potential relationship between existing entities and discovering new relational facts [6]. For giving a comprehensive understanding of the task of knowledge graph completion, this paper conducts a survey from the aspects of problem definition, main methods, evaluation indexes, etc. Meanwhile, the paper also analyzes the problems and challenges faced by knowledge graph completion task, and points out the future research directions of this field. The subsequent sections are organized as follows: In Section II, the introduction of knowledge graph completion is given. Section III introduces the main methods of knowledge graph completion. In this section, different knowledge graph completion technologies are reviewed, including their advantages, disadvantages and applicable fields. In Section IV and V, the main problems faced by knowledge graph completion and its applications are discussed. Section VI introduces the potential research directions of knowledge graph completion.

II. INTRODUCTION TO KNOWLEDGE GRAPH COMPLETION

A. GENERAL INTRODUCTION TO KNOWLEDGE GRAPH

Knowledge Graphs can be considered as an intelligent system integrating knowledge and data on a large scale, which can be traced back to the expert system developed in the late 1960s [7]. Under the background of combining knowledge with data, researchers try to construct knowledge-based system. In the early practice of modern knowledge graph construction, the knowledge (especially rules) of experts in a specific field often be coded. The Semantic Web appeared in the 1980s with the purpose of modeling the relationship between concepts. Berners-Lee et al. proposed to use the Se-

mantic Web to form link data and many high-quality knowledge bases have been built [8]. The current study originates from Google Knowledge Graph [9]. After that, companies such as Amazon, eBay, Facebook, IBM, LinkedIn and Uber have built the knowledge graph of their own industries.

The classification of knowledge graphs can be considered from the following perspectives: according to the modalities (content forms) of the graphs, they can be divided into text knowledge graphs, visual knowledge graphs, and multi-modality knowledge graphs [10]. In terms of domain scope, it can be divided into general knowledge graph and domain knowledge graph [7]. According to the timeliness of the contained knowledge, it can be divided into dynamic knowledge graph and static knowledge graph.

At present, the research on knowledge graph mainly focuses on three aspects: knowledge representation, knowledge graph construction, and knowledge graph application, which integrates cognitive computing, knowledge representation and reasoning, information retrieval and extraction, natural language processing, data mining and other technologies [1]. Knowledge graph completion is a hot topic in the field of knowledge graph construction and application. And this paper reviews the development in this field during the last ten booming years.

B. BASIC CONCEPTION OF KNOWLEDGE GRAPH COMPLETION

Since most knowledge graphs are constructed manually or semi-automatically, a large number of implicit entities and relationships have not been discovered, and therefore the incompleteness becomes a universal problem in almost all knowledge graphs. The goal of KGC is to solve the problems of incompleteness and sparsity caused by missing instances or links in the knowledge graphs, which is a necessary means to improve the quality of knowledge graphs. Knowledge graph completion technology complements the graph structure by predicting knowledge instances (entities, relationships, attributes, etc.), mining missing entities, relationships, or discovering new facts. It is an important means of discovering new knowledge and is widely applied in upper-level tasks of knowledge graphs (see Section IV-B).

For the knowledge graphs represented by Resource Description Framework (RDF), the triples like "head entity-relationship-tail entity" or "entities-attributes-attribute values" are used to describe nodes, edges and attributes in a graph network, in which the node corresponds to the entity in the real world, and edge represents all kinds of relations between entities. In this way, the knowledge graph completion problem can be converted into estimating the missing parts of the triples by using the methods like semantic similarity. According to the missing parts in triples, knowledge graph completion can be divided into three kinds of specific tasks: 1) given the head entities and relationships in a triples, predict the corresponding tail entities, such as (Beijing, capitalOf, ?) ; 2) given the relationship and tail entities, predict the corresponding head entities, such as (?, capitalOf, China); 3) given the head and tail entities, and predict the relationship between them, such as (China, ?, country). That is, from any two given elements in a triple and the third element can be deduced. For specific application, knowledge graph completion includes link prediction [11], [12], [13], entity prediction [14], [15], [16], relation prediction [17], attribute prediction [18] and other sub-tasks.

C. THE CLASSIFICATION OF KNOWLEDGE GRAPH COMPLETION

According to different task scenarios, knowledge graph completion can be divided into closed environment knowledge graph completion and open environment knowledge graph completion [19]. Specifically, if the entities and relationships involved in the completion process belong to the original knowledge graph, it is called the knowledge graph completion in the closed environment, also called the static knowledge graph completion; otherwise, it is called the knowledge graph completion in the open environment, also called dynamic knowledge graph completion.

At present, a large number of existing knowledge graph completion models are based on the closed environment hypothesis [20]. In such cases, all entities and relationships are supposed to be existing in the same knowledge graph, and graph completion can only be achieved by mining the potential connections between existing entities, instead of adding new entities and related relationships to the existing graph. The knowledge graph completion in closed environment relies heavily on the existing connected structure of the knowledge graph, which cannot achieve prediction for the weak connections and new entities and also cannot expand graph structure well. So the knowledge graph completion in closed environment is mostly applicable to the domain knowledge graph with small scale and slow update. And the KGC under closed environment does not make full use of external data for missing completion, resulting in insufficient information and strong limitations in usage.

Based on the assumption of open environment in probabilistic database theory [9], the knowledge graph completion model in open environment provides a method to predict external entities and weakly connected entities. Most of the existing large-scale knowledge graphs are constantly updated and expanded through linking external entities to adapt to the explosive growth of information. The knowledge graph completion in an open environment is relatively difficult to establish a connection between the local knowledge graph and the outside world, due to the wide range of alternative knowledge. But it has more advantages when expanding the scale of the knowledge graph [21]. Based on the knowledge graph completion in an open environment, the research in the field of dynamic updating of the knowledge graph has started. The existing research results [22], [23], [24] are carried out from the perspective of dynamic updating based on the data model layer and the data layer or adopting active learning strategies, which are beyond the scope of this review, so they will not be included.

Since most of the current research focuses on knowledge graph completion in closed environment, the contents of this paper are limited to this scope. According to the development of the research, the traditional knowledge graph completion methods and the others based on deep representation learning are introduced in turn. The former mainly includes rule-based reasoning method, probability graph model method, and the method based on graph computing. The latter includes translation model based method, semantic matching model based method, network representation learning method and neural network model based method.

D. METHODOLOGY

The current reviews [1], [8], [10], [25], [26], [27], [28], [29], [30] related to knowledge graphs mainly involve research on knowledge representation, knowledge graph construction, knowledge reasoning, etc., lacking of systematic review articles of knowledge graph completion. Literature [31] analyzes the graph completion algorithm based on knowledge representation, but it lacks comprehensiveness. Starting from the definition of the knowledge graph completion problem, this paper classifies the existing methods, and analyzes the advantages and disadvantages of each class, points out applicable fields of each type of method, introduces the evaluation indicators of the knowledge graph completion algorithms, and discusses the main challenges and problems in the field, as well as its potential research directions. Compared with the existing reviews, this article is more comprehensive and in-depth, reflects the latest research status, and can provides useful references for researchers in this field.

The selected documents are closely related to knowledge graph completion. These documents are taken from international journals in computer science and engineering technology indexed by EI/SCI and international conferences related to natural language processing, knowledge engineering, semantic web, and artificial intelligence, such as EMNLP, ACL, IJCAI, etc., which covers the content from related theories and model prototypes proposed in 1960s to the latest research methods in recent years. There are 120 articles covering about 80% of the literature in the last decade. These articles completely describe the development process of knowledge graph completion theory and related technologies, and depicts the research track in this field.

III. THE MAIN METHODS OF KNOWLEDGE GRAPH COMPLETION

According to the evolution of the knowledge graph complement methods, they are divided into two categories, namely the traditional knowledge graph complement and the methods based on knowledge representation learning. They are introduced below.

A. TRADITIONAL KNOWLEDGE GRAPH COMPLETION METHODS

1) Knowledge graph completion based on rule reasoning

The knowledge graph completion method based on rule reasoning uses rules or statistical features to deduce new knowledge to expand the graph structure and complete the knowledge graph [27]. For example, if there is a set *C* and its subset *C1*, according to the implicit rules, it can be deduced that the elements in the set *C* and subset *C1* belong to the same category. That is, the triple *(C1, equivalentClass, C)* can be reasoned from the rules by triple *(C1, subClassOf, C)*. NELL [32] is one of the earliest methods for rule-based knowledge reasoning. For a knowledge graph that defines ontology and its instances, NELL uses a first-order relational learning algorithm to deduce new relational instances from existing knowledge. Based on the semantic network inference rule OWL2RL, the knowledge graph completion model KGRL [33] effectively infers the hidden information in knowledge graph, and performs well in the prediction of discontinuous data.

Rule-based single-step reasoning methods rely on a large number of effective and accurate rules and statistical features. But effective and widely-covered rules and constraints are difficult to obtain, resulting in a low recall rate in the reasoning, so related research tends to the direction of multi-hop reasoning. Wang et al. proposed a probabilistic language model ProPPR [34], [35] which joins triples into clauses, takes clauses as inference targets and relation (edges) as an inference step, then associates the weight of the edge with the feature vector and adds self-loop from the target tail node to the start node to increase the weight of reasoning with fewer steps. Literature [36] proposes a reasoning completion based on an open knowledge graph, and proposes a novel collaborative policy learning (CPL) framework, which uses new facts extracted from a text corpus to perform multi-hop reasoning. The new facts augment the graph dynamically while performing reasoning.

For other graph-based and translation model-based multi-hop inference methods, see Section 2.1.3 and Section 2.2.1. In order to make full use of the rich semantic information contained in the knowledge graph, Paulheim and Bizer proposed algorithms SDType and SDValidate [37], which use the statistical distribution of attributes and types to complete the triple type. Semantic information is valuable feature information contained in the knowledge graph, which has been widely used in various methods of knowledge graph completion, and will be described in detail in the following sections.

Since rules are automatically generated according to semantics or extracted manually, the advantage of rule-based reasoning knowledge graph completion method lies in its strong interpretability. When comprehensive and accurate rules are obtained, the graph completion accuracy is high. At the same time, this method also has many defects. First of all, this method has a strong dependence on rules, and it is very difficult to obtain complete rules no matter through manual construction or automatic generation. Therefore, the expected reasoning accuracy and completion effect cannot

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

be achieved in practice. Secondly, the rule-based reasoning knowledge graph completion method is very inefficient in computation, especially in the condition that the scale of knowledge graph is increasing day by day, this traditional graph completion method can no longer meet the current application needs. Therefore, the knowledge graph completion method based on reasoning began to develop in the direction of distributed representation based reasoning, neural network based reasoning, and mixed reasoning.

2) Knowledge Graph Completion Method Based on Probabilistic Graph Model

Probabilistic graph model is a model that uses graphs to represent probabilistic relations, which provides the possibility to effectively reason and learn various types of probabilistic models [38]. The probabilistic graph model can combine the expression and computing power of graph with probability theory, so that it has the advantages of flexible topological structure, easy to understand and explain, obvious semantics and effective multi-information fusion in dealing with uncertainty problems [39]. Another reason why the probabilistic graph model is used in knowledge graph reasoning is its advantage in relational semantic interpretability. Not only is the reasoning process of the algorithm explicable, but also has the function of semantic interpretation [40]. Many researchers have also attempted to improve the semantic representation ability of probabilistic graphical models [41].

The knowledge graph completion method based on the probability graph model mostly uses Markov logic network [42] and Bayesian network [43]. Markov logic network is a probability distribution model with Markovian random variables. Combining first-order logic and probability graph models, the probability distribution of the knowledge graph ontology is derived according to the principle of large weights corresponding to strong rules. Then the Markov logic network is used to maximize its probability distribution, that is, the reasoning of knowledge is carried out through the principle of large weights corresponding to hard constraints [27]. Jiang et al. [44] proposed to extract first-order logic from ontology as hard constraint and extract weighted confidence from the instance ternary as soft constraint. And this Markov logic network combines hard constraint and soft constraint with probability graph model and can be used as a means to complete the knowledge graph. This kind of method makes link prediction using the potential characteristics of knowledge graph defined by logical rules. The bottleneck of its performance lies in rule learning and parameter estimation.

Bayesian network takes the network structure and node attribute information into consideration. It is a Directed Acyclic Graph (DAG) in form, with a solid theoretical foundation and wide application [45], suitable for expressing and analyzing uncertain knowledge and effective reasoning. It is one of the most effective models in the field of knowledge expression and reasoning [46]. Han [47] et al. proposed a knowledge graph completion model based on Bayesian network, in which a constructed knowledge graph was used to describe user interests. Combined with external data sets, as a representation and reasoning framework Bayesian network was used to measure the similarity and uncertainty among different commodities. And then, based on the probability reasoning mechanism, authenticity of the connection between commodity node and user node was quantitatively judged. Literature [39] combines Bayesian inference theory with latent factor model to realize link prediction, and judges the reliability of the relationship by calculating the confidence of the relationship between nodes, and discovers the potential relationship between entities. The model also considers the rich semantic information of the knowledge graph, adds annotation information to the knowledge graph node, and uses the ontology database to construct inference rules to complete the prediction. At the same time, it performs well in terms of improving the prediction accuracy and reducing time overhead.

Models based on probability graphs use joint probability distribution reasoning to predict new facts to complete the graph. Compared with rule-based methods, they improve the computation efficiency. However, due to the high complexity of the algorithm, it is difficult to reason and calculate for large-scale multi-relation knowledge graphs.

3) Knowledge graph completion method based on graph calculation

In the knowledge graph completion model based on graph calculation the structure of knowledge graph is abstracted as graph. Given a knowledge graph G, the graph structure G is a set of nodes and edges, and $G=(V, E)$, in which $v \in V$ is the node of graph, $e \in E \subseteq V \times v$ is the edge of graph, and e_{ij} is a connection edge between two nodes v_i and v_j. That is, the entity is expressed as the node in the graph, and the relationship of different types acts as the edges in the graph. By using different statistical characteristics of nodes and edges, such as the outgoing degree and incoming degree of nodes, adjacent matrix and so on, new entities and relationships can be predicted. Path Ranking Algorithm (PRA) [48] is one of the earlier graph completion methods based on graphs. Its basic idea is that for a given type of relation r, PRA uses Random Walk [49] method to acquire the specific path of knowledge graph, namely the specific relationship between entities r, and then uses the obtained path characteristic to train model. For a given set of entities, the trained model can be used to judge whether there is relationship r between them, that is, whether it can be connected by edge r, and if it can be, the triple will be added to the knowledge graph as a missing fact.

The path ranking algorithm is highly interpretable and does not require additional logic rules to assist in reasoning process. On the basis of this, Wang et al. proposed the Coupled Path Ranking Algorithm (CPRA) [50] model which use the similarity measure to collectively model the specific relationships that may have a common path, and divide the relationships into different groups. By this way, it improved the one-to-one modeling method of PRA for training a separate

classifier for each relationship. The graph-based knowledge graph completion method has three problems as follows: Firstly, the scalability is poor and the memory usage is high, because for a group of entity pairs, this type of algorithm requires enumerating paths to determine whether there exists all possible relationships between the entity pairs. Secondly, the number of paths is large, and using the path as a model training feature may cause feature explosion. Finally, like the completion method based on the probability graph model, the graph-based model is also facing the problem of high complexity of large scale data computation.

Traditional knowledge graph completion methods apply the reasoning rules and the network structure of knowledge graph. With the expansion of knowledge graph, the defects of this kind of method are gradually manifest. Firstly, the expansion of knowledge graph gradually reflects the sparsity of data, increases the difficulty of extracting rules, and long tail distribution entities associated knowledge is less. So the above methods are greatly limited in the aspect of knowledge graph completion; Secondly, the essence of knowledge graph data is a kind of semantic network, in which entities and relationships contain rich semantic information. However, it is difficult to obtain high-quality knowledge graph because the traditional knowledge graph representation methods cannot encode semantic information. Finally, the traditional knowledge graph completion method has the problem of computational efficiency, high algorithm complexity, poor portability and scalability. Based on this, the study of knowledge graph completion has shifted to the stage of knowledge representation learning.

B. MAIN METHODS OF KNOWLEDGE GRAPH COMPLETION BASED ON REPRESENTATION LEARNING

Because the knowledge graph is a multi-relational graph composed of entities (nodes) and relationships (different types of edges), it is usually organized in the form of a network. For example, the knowledge graph stored based on resource description framework (RDF) [51] is represented in triples. However, the knowledge graph representation based on network exists lots of problems in application, mainly including the following two aspects: First, the calculation efficiency. In the knowledge representation based on network graph, entities are expressed as different nodes. When calculating the semantic or reasoning the relationships between entities, it is necessary for specific application to design special graph algorithm to implement this representation. This method is poor in flexibility and scalability, which is difficult to meet the demand of the current large-scale knowledge graph calculation. Second, data sparsity problem. Similar to other types of large-scale data, large-scale knowledge graphs also follow long-tail distribution. The entities and relationships of the long-tail distribution face serious data sparsity problem [28]. For this problem, extensive attention has been turned to knowledge representation learning [52], [53], [54], [55], [56], [57] in recent years. Through machine learning, knowledge representation learning aims to express semantic information such as entities and relationships as dense low-dimensional real value vectors in a continuous vector space, which not only preserves the inherent graph structure of knowledge graph, but also simplifies operations. Typical knowledge representation learning techniques generally include the following three parts: 1) Represent relationships and entities in a continuous space; 2) Define the score function $f_r(h,t)$ to judge the probability of the establishment of triples *(h, r, t)*. The main difference between models lies in the difference of the score function; 3) Learn the representation of entities and relationships, and solve the optimization problem of maximizing the rationality of visible facts. Through the efficient computation of semantic relations between entities and relationships in low-dimensional space, the problem of data sparsity is effectively solved, and the effect of knowledge graph completion is significantly improved. The following will introduce the knowledge graph completion methods based on different representation learning models.

1) Knowledge Graph Completion Method Based on Translation Model

Translation model is the most representative classical method in knowledge representation learning. In 2013, Mikolov et al. proposed Word2Vec [58] algorithm for the first time, and thus proposed the translation invariant phenomenon of word vector, such as titanic-leonardodicaprio $\approx$ 2012-johncusack, that is, distribution based word representation captures some kind of same semantic relationship. According to the translation invariance phenomenon, Bordes et al. proposed the most representative classical translation model TransE [59], and led a large number of researchers into the study of Trans series models, in which the representative improved models include TransH [60], TransR [6] and TransD [61]. The main idea behind the translation model is to treat the process of finding valid triples as the translation operation of entities through relationships, define the corresponding score function, and then minimize the loss function to learn the representation of entities and relationships.

Given a training set S consisting of triples *(h, r, t)*, in the head and tail entity h, $t \in E$, E is entity set, and $r \in R$, R is relationship set. The main idea of TransE is that, if the triplet *(h, r, t)* is true, then the sum of the vector representations of head entity and relation is close to the vector representations of the tail entity; otherwise, it is far away, that is, when the triplet is formed, $\mathbf{h}+\mathbf{r}\approx\mathbf{t}$, as shown in FIGURE 1. From the above ideas, the score function $f_r(\mathbf{h},\mathbf{t}) = -\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{\frac{1}{2}}$ [59] of the TransE model can be obtained, which represents the Euclidean distance between the head entity and the tail entity in low-dimensional continuous space.

TransE model is efficient, concise and has good prediction effect, but there are two problems: 1) TransE uses the Euclidean distance as the distance metric in the score function, and each feature vector is assigned the same weight in calculation. And this makes the flexibility of the method poor, and the accuracy of knowledge representation may be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

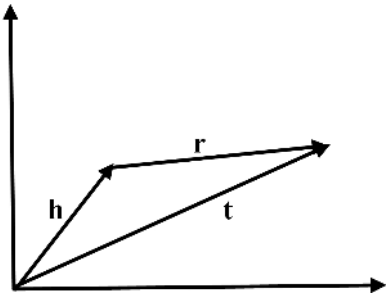

FIGURE 1: TransE model. [69]

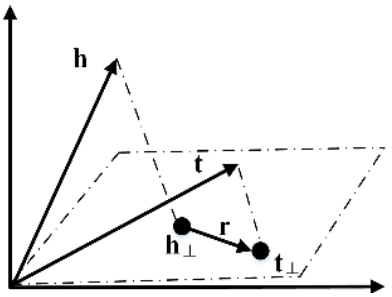

FIGURE 2: TransH model. [60]

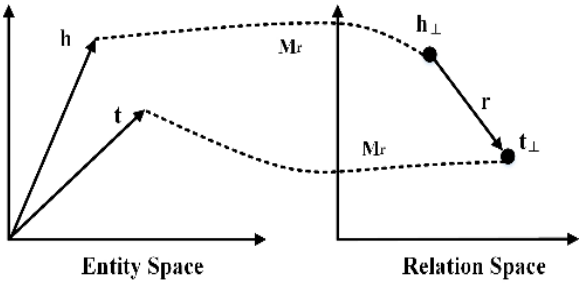

FIGURE 3: TransR model. [6]

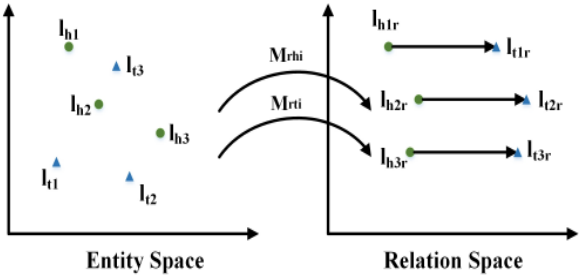

FIGURE 4: TransD model. [61]

affected by irrelevant dimensions; 2) The TransE model is relatively simple and has limitations in dealing with complex relationships such as reflexive, one-to-many, many-to-one, and many-to-many [6], [60], which cannot well distinguish entities with the same relationship [62].

Zhen et al. proposed model TransH to improve the above defects, embeds knowledge into the hyperplane of a specific relationship [60], as shown in FIGURE 2. TransH learns an additional mapping vector $\mathbf{W_r}$ for each relationship, which is used to map entities to the hyperplane specific to the relationship [27]. That is, for a triple *(h, r, t)*, the representation of the head and tail entities is first mapped to the hyperplane, and we can get $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w_r}^\top \mathbf{h} \mathbf{w_r}$, $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w_r}^\top \mathbf{t} \mathbf{w_r}$ [60]. If the triple is true, the relationship vector $\mathbf{r}$ on the hyperplane can be used to connect the head-to-tail vector mapped to this hyperplane. At this time on the hyperplane $f_r(\mathbf{h},\mathbf{t}) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2$ [60]. To some extent, TransH model alleviates the problem that TransE model cannot handle complex relationships well. The TransR model [6] represents entities and relationships in separate entity and relationship spaces according to specific relationships, as shown in FIGURE 3. That is, for a triple *(h, r, t)*, the representation of the head and tail entities is first mapped to the space corresponding to a specific relationship, and gets $\mathbf{h_1} = \mathbf{M_r h}$, $\mathbf{t_1} = \mathbf{M_r t}$ [8] . If the triple is established, the relationship vector is regarded as the transfer between entity vectors in the corresponding space. The score function is defined as $f_r(\mathbf{h},\mathbf{t}) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2$ [8]. Although the TransR model has some improvements compared to the original translation model, it still has the following problems: First, head and tail entities connected by the same relationship may differ greatly in type or attribute, which will have a certain impact on prediction accuracy. Second, the projection matrix in TransR is formed according to different relationships, ignoring the impact of different types of entities. Third, TransR, while introducing the projection matrix, increases the number of parameters and computation complexity. The literature [61] proposed the TransD model to further improve the TransR model by introducing two projection vectors $\mathbf{M_{rh}}$ and $\mathbf{M_{rt}}$ to represent the mapping of head entities and tail entities to the relationship space. As shown in FIGURE 4, the projection vector is defined as Equation (1).

$$\begin{aligned} \mathbf{M_{rh}} &= \mathbf{l_{rp}} \mathbf{l_{hp}} + \mathbf{I}^{d \times k}, \\ \mathbf{M_{rt}} &= \mathbf{l_{rp}} \mathbf{l_{tp}} + \mathbf{I}^{d \times k}. \end{aligned} \tag{1}$$

TransD replaces the operation of matrix and vector multiplication in the previous model with the operation of vector multiplication, which improves the calculation efficiency to some extent and solves the problem of too many parameters in the TransR model. So TransD is suitable for the completion of large-scale knowledge graphs. Another improved version of TransR is TranSparse model proposed by Ji et al. [63], which replaces dense matrix in the original model with sparse matrix as mapping matrix, solves the distribution imbalance of entities and relationships in knowledge graph, and reduces the number of parameters in the model at the same time. For the head and tail entities connected by the same relationship, two different sparse mapping matrices are introduced, $\mathbf{M_1^r}(\theta_1^r)$ and $\mathbf{M_2^r}(\theta_2^r)$, that is $\mathbf{h}_\perp = \mathbf{M_r}(\theta_r)\mathbf{h}$, $\mathbf{t}_\perp = \mathbf{M_r}(\theta_r)\mathbf{t}$ [40], in which θ_r, θ_1^r, θ_2^r denotes the sparsity of the matrix, defined as $\theta_r^1 = \frac{1-(1-\theta_{min}N_r^1)}{N_r^1}$, where N represents the number of connected entities of relation r at position 1. TranSparse model score function is defined as $f_r(\mathbf{h},\mathbf{t}) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2$.

The knowledge graph completion based on translation model predicts new entity relationship triples from the existing knowledge graph through embedding entities and relationships in the vector space. Most of the existing methods focus on the structured information of triples and maximize the possibility of their establishment [64], but they ignore the semantic information and the prior knowledge indicated by semantic information contained in most knowledge graphs. Considering encoding semantic information, literature [64] proposed a method TransT that integrates structured information and entity types describing entity categories. The relationship type is constructed from the entity type, and the type-based semantic similarity of related entities and relationships is used to capture the prior distributions of entities and relationships. But this method ignores the semantic information contained in most knowledge graphs and the prior information indicated by the semantic information, using type-based prior distributions, TransT generates multiple embedding representations of each entity in different contexts, and estimates the posterior probability of entity and relationship predictions. Literature [65] proposed a generative model TransG to solve the problem of multi-relational semantics, that is, a relationship may have multiple meanings for different entity pairs associated in different triples. It is proposed to use a Gaussian mixture model to describe the relationship between head and tail entities. Each semantic is described by a Gaussian distribution. See Equation (2).

$$r = \sum_i \pi_r^i \mu_r^i, \mu_r^i \sim N(\mu_t - \mu_h, (\sigma_h^2 + \sigma_t^2)\mathbf{I}). \tag{2}$$

The score function definition of this model is shown in Equation (3).

$$\begin{aligned} P\left\{(\mathbf{h},\mathbf{r},\mathbf{t})\right\} &\propto \sum_{m=1}^{M_r} \pi_{r,m} P(\mathbf{u}_{\mathbf{r},\mathbf{m}}|\mathbf{h},\mathbf{t}) \\ &= \sum_{m=1}^{M_r} \pi_{r,m} e^{-\frac{\left\|\mathbf{u_h}+\mathbf{u}_{\mathbf{r},\mathbf{m}}-\mathbf{u_t}\right\|_2^2}{\sigma_h^2+\sigma_t^2}}. \end{aligned} \tag{3}$$

TransG can discover the underlying semantics of relations and embed triples using the combination of relation-specific component vectors. This is the first generative model for knowledge graph embedding, and the issue of multiple relational semantics is formally discussed for the first time. The literature [66] comprehensively considered the above-mentioned method to model the relationship, and proposed the RotatE model from the perspective of inferring the relationship synthesis mode. This method can model and infer various relationship modes, including: symmetric/antisymmetric, inversion, and composition. The RotatE model defines each relationship as the rotation from the source entity to the target entity in the complex vector space. The proposed RotatE model is not only scalable and linear in time and memory, so it can be applied to large-scale knowledge graph; and it can also infer and model various relational models, which is significantly better than the existing models for link prediction.

Table 1 summarizes the characteristics of Trans series models.

The knowledge graph completion method based on the translation model focuses on the use of the relationship between entities, the semantics contained in the entity and relationship, and the structured information of the knowledge graph to realize the modeling of entities and relationships, which makes up for the complex training and difficult extension of traditional methods. For modeling entities and relationships, the methods are very simple and clear with strong interpretability [59].

2) Knowledge Graph Completion Method Based on Semantic Matching Model

i)RESCAL and its extension

The semantic matching model uses the score function based on semantic similarity to mine the potential semantic association between entities and relationships, and by embedding the representation of entities and relationships in the vector space, it can obtain the possibility of new facts, so as to predict the new knowledge and complete the knowledge graph. The following are some classical representation methods based on semantic matching:

For the representation method based on bilinear model [70], relations after bilinear transformation is used to describe the semantic connection between entity and relation, and to capture various interactions between data. The most classical representative method is the RESCAL model [52]. As a kind of relational data, knowledge graph is characterized by the correlation between multiple interconnected nodes, such as entity attribute categories and other hidden semantic information. RESCAL is a relational learning method based on three-way tensor factorization, in which each entity is associated with a vector to capture its implied semantics. The relationship matrix is used to modeling interactions between latent factors. The RESCAL model is shown in FIGURE 5. The score of triples *(h, r, t)* is defined by the bilinear function as Equation (4):

$$f_r(\mathbf{h},\mathbf{t}) = \mathbf{h}^\top \mathbf{M_r} \mathbf{t} = \sum_{i=0}^{d-1}\sum_{i=0}^{d-1} [\mathbf{M_r}]_{ij} \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_j, \tag{4}$$

where $\mathbf{h}$ and $\mathbf{t}$ are the vector representations of the entities and $\mathbf{M_r}$ is the relation matrix. The score function captures all interactions between pairs of head and tail entities.

Ji et al. proposed the DistMult model [70] shown in FIGURE 6, which is a simplification of the RESCAL model, and decomposes the relation matrix $\mathbf{M_r}$ into a diagonal matrix. For each relation $\mathbf{r}$ in the vector space, make $\mathbf{M_r} = \mathrm{diag}(\mathbf{r})$ [70]. Then its score function becomes Equation (5):

$$f_r(\mathbf{h},\mathbf{t}) = \mathbf{h}^\top \mathrm{diag}(\mathbf{r})\, \mathbf{t} = \sum_{i=0}^{d-1} [\mathbf{r}]_i \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_i. \tag{5}$$

The score function captures the underlying relationships between pairs of head-tail entities that are only in the same

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

TABLE 1: The Characteristics of Trans Series Models

Method	Score function	Constrains/Regularization	Characteristics
TransE [59]	$f_r(\mathbf{h},\mathbf{t}) = -\left\|\mathbf{h}+\mathbf{r}-\mathbf{t}\right\|_{\frac{1}{2}}$	$\left\|\mathbf{h}\right\|_2 = 1, \left\|\mathbf{t}\right\|_2 = 1$	The embedding of multiple relationships
TransH [27]	$-\left\|(\mathbf{h}-\mathbf{w_r}^\top\mathbf{h}\mathbf{w_r})+\mathbf{r}-(\mathbf{t}-\mathbf{w_r}^\top\mathbf{t}\mathbf{w_r})\right\|_2^2$	$\left\|\mathbf{h}\right\|_2 \leqslant 1, \left\|\mathbf{t}\right\|_2 \leqslant 1$	Mapping entities to hyperplanes corresponding to specific relationships
TransR [6]	$f_r(\mathbf{h},\mathbf{t}) = -\left\|\mathbf{h}_\perp+\mathbf{r}-\mathbf{t}_\perp\right\|_2^2$	$\left\|\mathbf{w_r}^\top\right\| / \left\|\mathbf{r}\right\|_2 \leqslant \varepsilon, \left\|\mathbf{w_r}\right\|_2 = 1$ $\left\|\mathbf{h}\right\|_2 \leqslant 1, \left\|\mathbf{t}\right\|_2 \leqslant 1 \left\|\mathbf{r}\right\|_2 \leqslant 1$	Mapping different relationships to different semantic spaces
TransA [67]	$-(\left\|\mathbf{h}+\mathbf{r}-\mathbf{t}\right\|)^\top\mathbf{M_r}(\left\|\mathbf{h}+\mathbf{r}-\mathbf{t}\right\|)$	$\left\|\mathbf{h}\right\|_2 \leqslant 1, \left\|\mathbf{t}\right\|_2 \leqslant 1 \left\|\mathbf{r}\right\|_2 \leqslant 1$	Solving the problem of over-simplification of loss measurement in translation-based knowledge representation methods and lack of competitiveness to measure the diversity and complexity of entities/relationships in the knowledge base
TransD [61]	$-\left\|(\mathbf{w_r}\mathbf{w_h}^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w_r}\mathbf{w_t}^\top+\mathbf{I})\mathbf{t}\right\|_2^2$	$\left\|\mathbf{M_r}\mathbf{h}\right\|_2 \leq 1, \left\|\mathbf{M_r}\mathbf{t}\right\|_2 \leq 1$ $\left\|\mathbf{h}\right\|_2 \leqslant 1, \left\|\mathbf{t}\right\|_2 \leqslant 1 \left\|\mathbf{r}\right\|_2 \leqslant 1$ $\left\|(\mathbf{w_r}\mathbf{w_h}^\top+\mathbf{I})\mathbf{h}\right\|_2 \leq 1$ $\left\|(\mathbf{w_r}\mathbf{w_t}^\top+\mathbf{I})\mathbf{t}\right\|_2 \leq 1$	Using different vectors for head and tail entities to represent their projections in the vector space
TranSparse [63]	$-\left\|\mathbf{M_r}(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M_r}(\theta_r)\mathbf{t}\right\|_{1/2}^2$ $-\left\|\mathbf{M_r^1}(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M_r^2}(\theta_r^2)\mathbf{t}\right\|_{1/2}^2$	$\left\|\mathbf{h}\right\|_2 \leqslant 1, \left\|\mathbf{t}\right\|_2 \leqslant 1$ $\left\|\mathbf{M_r}(\theta_r)\mathbf{h}\right\|_2 \leqslant 1$ $\left\|\mathbf{M_r}(\theta_r)\mathbf{t}\right\|_2 \leqslant 1$ $\left\|\mathbf{M_r^1}(\theta_r^1)\mathbf{h}\right\|_2 \leqslant 1$ $\left\|\mathbf{M_r^2}(\theta_r^2)\mathbf{t}\right\|_2 \leqslant 1$	For the relationship between entities of different difficulties, using matrices with different degrees of sparseness to represent, to prevent under-fitting complex relationships or over-fitting simple relationships
TransT [64]	$\sum_{(h,r,t)\in\Delta}\sum_{(h',r',t')\in\Delta'_{(h,r,t)}}$ $\max\left\{0,\gamma+l(h,r,t,h',r',t')\right\} =$	$l(h,r,t,h',r',t') =$ $-lnp(h\|r,t,true)+$ $lnp(h'\|r,t,true), h' \neq h$ $-lnp(t\|h,r,true)+$ $lnp(t'\|h,r,true), t' \neq t$ $-lnp(r\|h,t,true)+$ $lnp(r'\|h,t,true), r' \neq r$	Predicting entities and relationships based on semantic information embedding of structured information and entity type information
TransG [68]	$\sum_i \pi_r^i \exp(-\frac{\left\|\mu_h+\mu_r^i-\mu_t\right\|_2^2}{\sigma_h^2+\sigma_t^2})$	$\left\|\mu_h\right\|_2 \leqslant 1, \left\|\mu_t\right\|_2 \leqslant 1$ $\left\|\mu_r^i\right\|_2 \leqslant 1$	Forming multiple Gaussian distributions based on the different semantics of a relationship to distinguish correct and incorrect entities
TransM [69]	$-\theta_r\left\|\mathbf{h}+\mathbf{r}-\mathbf{t}\right\|_{1/2}$	$\left\|\mathbf{h}\right\|_2 = 1, \left\|\mathbf{t}\right\|_2 = 1$	Using different weights to update parameters for different relationship types
RotatE [66]	$-\left\|\mathbf{h}\circ\mathbf{r}-\mathbf{t}\right\|^2$	$\mathbf{h},\mathbf{r},\mathbf{t} \in C^k, \left\|r_i\right\| = 1$	Defining each relationship as a rotation from the source entity to the target entity in the complex vector space

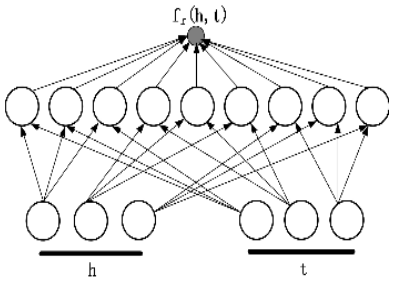

FIGURE 5: RESCAL model. [52]

dimension. Compared with RESCAL model, DistMult model reduces the number of parameters and the complexity of the model, and significantly improves the mining effects of the potential information in knowledge graph.

ComplEx [71] model is an extension of the DistMult model. The previous models for knowledge graph completion use three-dimensional bivariate tensor completion for link prediction. Each slice of tensor denotes the adjacency matrix of a certain relationship in the knowledge graph. The low-rank decomposition is carried out for the tensor, and each row of the decomposed matrix is used to represent an entity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

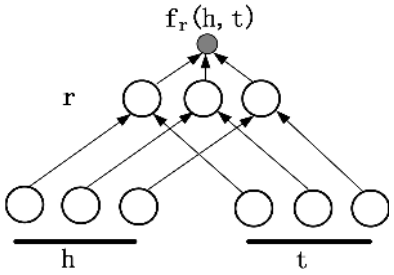

FIGURE 6: DistMult model. [70]

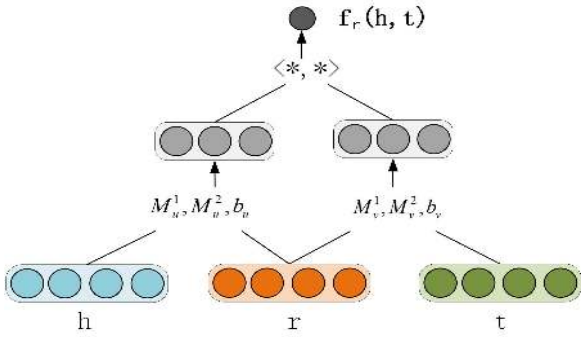

FIGURE 7: Semantic matching energy model. [80]

or a relationship of the knowledge graph. Finally, for a given triplet *(h, r, t)*, the score is calculated by the multi-linear product among the vector representations, and then decide whether the triplet can be linked to the graph structure. Such a method is not so suitable for the prediction of asymmetric relations, so ComplEx model is proposed to model asymmetric relations, that is, entities and relations are embedded into ComplEx Spaces, and the score function is defined as Equation (6):

$$f_r(\mathbf{h},\mathbf{t}) = Re\left(\mathbf{h}^{\top}\mathrm{diag}(\mathbf{r})\bar{\mathbf{t}}\right) = Re\left(\sum_{i=0}^{d-1}[\mathbf{r}]_i\cdot[\mathbf{h}]_i\cdot[\bar{\mathbf{t}}]_i\right). \quad (6)$$

This model can obtain good accuracy when predicting symmetric relationships, can well excavate potential semantic associations, and can achieve better results than the benchmark methods in semantic mining and prediction for asymmetric relationships.

In addition, there is HolE [72], another semantic matching model for knowledge graphs embedding, which combines RESCAL's expressive ability and DistMult's [25] efficiency and simplicity to learn the component vector space representation of the whole knowledge graph. The proposed method is related to the holographic model of associative memory because it employs cyclic correlation to create a representation of the composition. By using correlation as a composite operator, HolE can capture rich interactions, but still compute efficiently, train easily, and scale to very large data sets. In this paper [73], another extended ANALOGY model of RESCAL was put forward to further simulate the analogy properties of the entity and relationship, optimize their potential representation, and fill in the gap of predicting the new triples from the perspective of analogy reasoning.

The model QuatE [65] uses two rotating planes in the complex space to preserve the symmetry/antisymmetric, flip, and combination relations, and models the relations as rotations in a hyper-complex space, thus unifying ComplEx and RotatE. Compared with RotatE, QuatE requires 80% less free parameters for training on FB15k-237.

The knowledge graph completion methods based on semantic matching model and its characteristics are summarized in Table 2.

ii)Matching with Neural Networks

The above methods are mostly expressed in distributed forms of entity and relationship types, which have been shown to help improve the performance of standard natural language processing tasks (NLP) [76]. For multi-relational data, linear relation embedding [77], [78] imposes constraints, that is, through linear operations to model the relationship in this feature space, thereby learning the mapping from the entity to the feature space. This idea is further improved in the structured embedding (SE) framework proposed by Bordes et al. [79]. Semantic matching energy model (SME) [74] uses neural network structure to achieve semantic matching, mining semantic relationships between entities and relationships. The semantic matching energy model is proposed to learn the distributed representation of multi-relational data, which has the same trend as the above research. The relationship type is not modeled by matrix, but represented by vector, so the state and number of parameters of entities can be shared, which is suitable for the situation with a large number of relationship types. The semantic matching energy model (SME) can be divided into linear form and bilinear form. The linear form is defined in Equation (7).

$$\begin{aligned} G_u(\mathbf{h},\mathbf{r}) &= \mathbf{M_u^1 h} + \mathbf{M_u^2 r} + \mathbf{b_u}, \\ G_v(\mathbf{t},\mathbf{r}) &= \mathbf{M_v^1 t} + \mathbf{M_v^2 r} + \mathbf{b_v}. \end{aligned} \quad (7)$$

The bilinear form is defined in Equation (8):

$$\begin{aligned} G_u(\mathbf{h},\mathbf{r}) &= (\mathbf{M_1^u h}) \circ (\mathbf{M_2^u h}) + \mathbf{b_u}, \\ G_v(\mathbf{t},\mathbf{r}) &= (\mathbf{M_1^v t}) \circ (\mathbf{M_2^v r}) + \mathbf{b_v}, \end{aligned} \quad (8)$$

in which $\mathbf{M_1^u}$, $\mathbf{M_2^u}$, $\mathbf{M_1^v}$, $\mathbf{M_2^v}$ are weight matrix. The semantic matching energy model is shown in FIGURE 7.

Among the above two forms, the linear model uses simple linear layer and the bilinear model uses 3-modes tensors as core weights. The linear model represents a triplet as a combination of bigrams, while the bilinear model registers a triplet as a trigram. The bilinear SME model performs better than the linear model on the linked prediction task, but the former requires d times more parameters to learn than the latter [56] [80].

The neural tensor network (NTN) model [81] is one of the representative models. The basic idea is to replace the linear transformation layer in the traditional neural network

TABLE 2: Characteristics Based on Semantic Matching Model

Method	Score function	Constrains/Regularization	Characteristics
RESCAL [52]	$\mathbf{h}^\top \mathbf{M_r} \mathbf{t}$	$\|\mathbf{h}\|_2 \leqslant 1, \|\mathbf{t}\|_2 \leqslant 1,$ $\|\mathbf{M_r}\|_F \leqslant 1$ $\mathbf{M_r} = \sum_i \pi_r^i u_i v_i^\top$	Associate each entity with a vector to capture its latent semantics.
DistMult [25]	$\mathbf{h}^\top \mathrm{diag}(\mathbf{r}) \mathbf{t}$	$\|\mathbf{h}\|_2 = 1, \|\mathbf{t}\|_2 = 1, \|\mathbf{r}\|_2 \leqslant 1$	Mining logical rules with learned relational embedding
HolE [72]	$\mathbf{r}^\top (\mathbf{h} * \mathbf{t})$	$\|\mathbf{h}\|_2 \leqslant 1, \|\mathbf{t}\|_2 \leqslant 1, \|\mathbf{r}\|_2 \leqslant 1$	Using holographic embedding to learn the component vector space representation of the entire knowledge graph
ComplEx [71]	$Re(\mathbf{h}^\top \mathrm{diag}(\mathbf{r}) \bar{\mathbf{t}})$	$\|\mathbf{h}\|_2 \leqslant 1, \|\mathbf{t}\|_2 \leqslant 1, \|\mathbf{r}\|_2 \leqslant 1$	Using the composition of complex valued embedding to handle a large variety of binary relations
ANALOGY [73]	$\mathbf{h}^\top \mathbf{M_r} \mathbf{t}$	$\|\mathbf{h}\|_2 \leqslant 1, \|\mathbf{t}\|_2 \leqslant 1,$ $\|\mathbf{M_r}\|_F \leqslant 1$ $\mathbf{M_r} \mathbf{M_r}^\top = \mathbf{M_r}^\top \mathbf{M_r}$ $\mathbf{M_r} \mathbf{M_{r'}} = \mathbf{M_{r'}} \mathbf{M_r}$	Optimizing their potential representation with the analog attributes of embedded entities and relationships
SME [74]	$(\mathbf{M_u^1 h} + \mathbf{M_u^2 r} + \mathbf{b_u})^\top$ $(\mathbf{M_v^1 t} + \mathbf{M_v^2 r} + \mathbf{b_v})$ $((\mathbf{M_u^1 h}) \circ (\mathbf{M_u^2 r}) + \mathbf{b_u})^\top$ $((\mathbf{M_v^1 t}) \circ (\mathbf{M_v^2 r}) + \mathbf{b_v}))$	$\|\mathbf{h}\|_2 = 1, \|\mathbf{t}\|_2 = 1$	Encoding multiple relationship data into entities and relationships to learn the semantics of multiple relationships
QuatE [65]	$f_r(h,t) = \mathbf{h} \otimes \mathbf{r}^\triangleright \cdot \mathbf{t}$	$L(Q,W) = \sum_{r(h,t) \Omega \cup \Omega^-}$ $\log(1 + \exp(-Y_{hrt} \phi(h,r,t)))$ $\lambda_1 \|Q\|_2^2 + \lambda_2 \|W\|_2^2$	Providing a better spatial explanation, the extension of ComplEx in hypercomplex space
NTN [72]	$\mathbf{r}^\top tanh(\mathbf{h}^\top \mathbf{M_r} \mathbf{t} + \mathbf{M_r^1 h} + \mathbf{M_r^2 t} + \mathbf{b_r})$	$\|\mathbf{h}\|_2 \leqslant 1, \|\mathbf{t}\|_2 \leqslant 1, \|\mathbf{r}\|_2 \leqslant 1$ $\|\mathbf{b_r}\|_2 \leqslant 1, \left\|\mathbf{M_r^{[:,:,i]}}\right\|_F \leq 1$ $\left\|\mathbf{M_r^1}\right\|_F \leqslant 1, \left\|\mathbf{M_r^2}\right\|_F \leq 1$	High-precision prediction of hidden relationships between entities

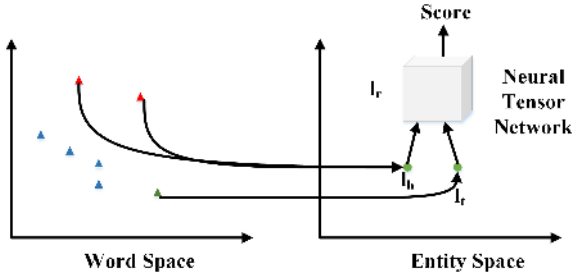

FIGURE 8: Neural tensor network model. [81]

with a bilinear tensor, and connect the head and tail entity vectors in different dimensions, as shown in FIGURE 8. Each relational triple is described by a neural network and a pair of entities as the input of the relational model. If entity pairs can be connected by this relationship, the model will return a high score. The score function reflects the possibility of a relationship between the two entities. The score function of the neural tensor network model is shown in Equation (9):

$$f_r(\mathbf{h}, \mathbf{t}) = \mathbf{u_T^t} g(\mathbf{l_h} \mathbf{M_r} \mathbf{l_t} + \mathbf{M_{r1}} \mathbf{l_h} + \mathbf{M_{r2}} \mathbf{l_t} + \mathbf{b_r}), \quad (9)$$

where $g()$ is the activation function, $\mathbf{u_T^t}$ is the linear layer related to the relationship, and $\mathbf{M_r}$ is the relation projection matrix. NTN can achieve high-precision prediction of hidden relationships between entities. The introduction of tensor can accurately describe the complex semantic relationship between entities and relationships. At the same time, there are problems with many model parameters and high computational complexity, which cannot be adapted to large-scale knowledge graphs completion.

Knowledge graph is multi-relational data and knowledge graph completion is a learning task for multi-relational data. Because the superposition of independent learning models specific to each relationship is extremely inefficient, especially because the relationships observed for each relationship are very sparse, there may be redundancy between entities. As an important multi-relationship learning algorithm, semantic matching model can capture the correlation between entities or relationships through their attributes, relationships or categories. The essence of knowledge graph completion task is how to apply observed triples in incomplete graph to predict whether the unobserved triples in graphs are established or not. This is a serious challenge for machine learning research. The above bilinear embedding model takes into consideration the latent semantic information of entities and relationships, and can obtain entities and relationships of deep mutual information, in order to better complete the knowledge graph task completion.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

3) Knowledge graph completion method based on network representation learning

As a kind of graph data, the valuable characteristics of knowledge graph not only include node (entity) and side (relationship), but also the attribute of nodes and edges and the characteristics of graph network structure. The study of these characteristics can help us exploit the potential characteristics in the knowledge graph and achieve higher accuracy of knowledge graph completion. The method based on network representation learning aims at fusing the information extracted from the network topology structure and the content information of nodes and edges, transforming the network vertices into the embedding representations in low-dimensional continuous vector space, and implementing the task of knowledge graph completion with the help of machine learning. This type of model was first proposed based on the DeepWalk model [82]. The purpose is to obtain the embedding representation of network nodes in a low-dimensional continuous vector space. For nodes that belong to the same adjacency structure or have similar functions (such as nodes with the same connection structure) can get similar embedding through learning.

The basic idea of the DeepWalk algorithm draws on the text generation process to construct a random walk path on the network [49], which is equivalent to a natural sentence, expressed as a sequence of nodes. Specifically, the algorithm uses the language model Skip-gram to maximize the co-occurrence probability of vertices appearing in a local structure, models the network representation by maximizing the co-occurrence probability of vertices in the same window [83], and finally uses stochastic gradient descent to learn the parameters. The advantage of the DeepWalk model is that multiple random walks can be performed at the same time to generate node sequence information, and the sequence generation relying only on local network information saves calculation time and storage space, and can avoid the uncertainty of adjacency matrix modeling. When graph structure changes, there is no need to recalculate the overall situation. Graphs can be updated iteratively. So the model can be applied to large-scale sparse graph data calculation. Its disadvantage is that the walk between nodes is random, ignoring the transition probability between nodes, which does not conform to the actual application scenario.

By changing the way of random walk sequence generation, Node2vec [84] further extends DeepWalk algorithm. Node2vec is a node vectorization model, and proposes a learning algorithm for vertices in the network structure, in order to get the embedding representation in the low-dimensional continuous vector space. In this way, similar embedding can be learned for nodes with the same adjacent structure and nodes with similar functions (such as nodes with the same connection structure).

In response to the shortcomings of the DeepWalk model mentioned above, Tang et al. proposed an improved model LINE [85]. LINE is suitable for undirected or directed graphs and can assign weight to different nodes and edges. The local

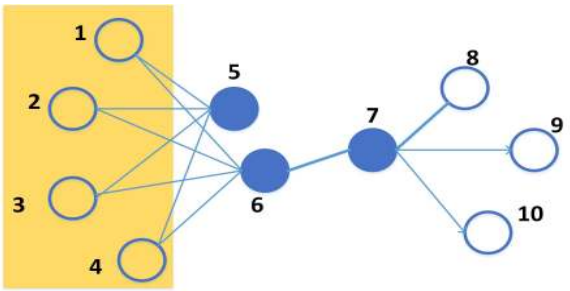

FIGURE 9: Network structure diagram. [85]

and global information of the network structure is preserved by first-order and second-order similarity of network nodes. First-order similarity refers to a pair of vertices that can be directly connected by edges. It represents the probability of one vertex pointing to another vertex. (First-order similarity is used in undirected graphs where two vertices point to each other with the same probability). If the two vertices cannot be directly connected, the first-order similarity between them is 0. Then weight is assigned based on the closeness of the relationship between the two vertices. The larger the weight, the higher the first-order similarity. Second-order similarity is used to describe the probability that there is an association between vertices with the same adjacency structure. If there are no identical neighbor vertices between two vertices, the second-order similarity is 0. As shown in FIGURE 9, vertices 6 and 7 are directly connected, and the similarity between them can be described by first-order similarity. Vertices 5 and 6 have the same adjacency network structure, and the similarity between them corresponds to second-order similarity.

The LINE algorithm selects KL(Kullback-Leibler) divergence as the distance function, and minimizes it to obtain the embedding representation of the vertex in the d-dimensional vector space, while retaining the first-order and second-order adjacency structure information of the vertex. Applying this method based on network representation to knowledge graph completion task can better extract the hidden features in the structure of the knowledge graph, which is beneficial to the training of the prediction model.

The above-mentioned networks are all shallow networks, and it is difficult to deal with highly nonlinear and sparse network structures. In response to this problem, SDNE [86] constructed a deep network composed of multiple nonlinear mapping layers, which can capture highly nonlinear network structure. SDNE consists of two parts, one is the module for modeling the first-level similarity supervised by the Laplace matrix, and the second is the module for modeling the second-level similarity relationship by the unsupervised deep autoencoder. In the end, the SDNE algorithm takes the middle layer of the deep autoencoder as the network representation of the nodes, and combines supervised and semi-supervised learning to obtain network expressions suitable for multiple tasks. The obtained expressions can simultaneously reflect the local and global structural information of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

network.

It is theoretically proved that the essence of DeepWalk, LINE and node2vec models [82] can be explained by implicit matrix factorization, and is inherently related to graph Laplace. Based on this understanding, NetMF [87] gave a general framework for decomposing closed matrices, which can achieve consistent performance improvement compared with models such as DeepWalk and LINE.

However, NetMF [87] has problems with computational efficiency when facing large-scale networks. For example, for a network with its node size of more than a dozen millions and a number of edges in a few hundred million, the requirements for computing resources are tens of times higher than other algorithms. The practicality is greatly restricted. In order to retain the performance of the NetMF algorithm and reduce the requirements for computing resources, NetSMF proposes to sparse the NetMF matrix [88], and keep the spectrum of the sparse matrix close to the original matrix, and then decompose the sparse matrix. Experiments show that this method can effectively improve the computational efficiency of the algorithm while retaining the spectrum information of the network. ProNE [89] further uses spectral propagation to enhance network embedding on the basis of sparse matrix decomposition, so that the learned embedding can not only capture the local structure information of the network, but also obtain the global network characteristics.

Asymmetric transitivity is an important attribute of directed graphs, from which the graph structure can be captured and graphs can be restored from partial observation graphs. Most of the existing graph representation learning methods are suitable for undirected graphs, and the graph structure and intrinsic properties can be learned by learning. However, for directed graphs, it is difficult to maintain the asymmetric transitivity of directed graphs. HOPE [90] assigns similar or different values to the source node vector (source vector) and target node vector (target vector) of the two nodes to maintain the relationship according to the directed link between the two nodes The asymmetry of [30], and the use of high-order approximate metric in embedding learning to obtain this result.

The above network representation learning algorithm is used for the link prediction task, and Area Under the Receiver Operating Characteristic Curve (ROC AUC) is used as the evaluation indicator. The performance evaluation results on the two real world networks PPI and Wikipedia are shown in Table 3[1].

TABLE 3: Performance comparison of network representation learning methods in link prediction tasks

Rank	Method	PPI	Wikipedia
1	DeepWalk [82]	69.65	65.93
2	LINE [85]	73.75	66.51
3	Node2vec [84]	70.19	66.60
4	SDNE [86]	54.87	60.72
5	Hope [90]	80.21	68.89
6	NetMF [87]	79.04	73.24
7	NetSMF [88]	68.64	67.52
8	ProNE [89]	79.93	82.74

4) Other knowledge graph completion methods based on neural network models

The knowledge graph completion method based on neural network applies strong learning and expression ability of neural network to model the knowledge graph, which can obtain good reasoning ability [27]. In recent years, the study of graph data analysis based on machine learning algorithms has attracted wide attention. As an efficient machine learning algorithm for non-Euclidean structural graph data, graph neural network model has achieved good results in node classification, link prediction, clustering and other applications. In the following part, the representative researches on the neural network based graph completion model are summarized.

i) Knowledge graph completion based on convolutional neural network

Deep neural network and representation learning provide new methods to solve the problem of data sparsity. Researchers have proposed a number of neural network models for learning word representations. Due to its efficient expression and efficiency, convolutional neural network is introduced into natural language processing tasks to complete traditional natural language processing tasks [91], sentence classification modeling [92] [93] and other tasks.

In literature [94], the convolutional neural network model ConvE was proposed to complete link prediction and knowledge graph completion. In the completion of a large knowledge graph, some shallow models are often used for link prediction task. But these kinds of model lack the ability to extract underlying deeper features, producing poor prediction effects. To enhance the ability of the models to extract features, the complexity and the number of parameters of the models is usually increase, the number of parameters is proportional to the number of entities and relationships. These methods cannot be used for large-scale knowledge graph, and the method of increasing neural network layer is likely to cause a problem such as overfitting [29]. In order to solve the contradiction between data scale and overfitting, literature [94] proposed ConvE, a 2D convolutional neural network model with high parameter efficiency and extendibility, to implement the representation learning of knowledge graph, and to predict new knowledge in knowledge graph by using 2D convolutional model based on representation. The model structure is shown in FIGURE 10. First the representation of entities and relationships will be refactored and connected, and then input the obtained matrix to convolution layer. The obtained characteristics mapping matrix will be projected into k-dimensional space for vectorization, and be matched with the vector representation of all candidate entities. Convolution neural network has the ability of extracting multi-scale local space feature and combining them to build effi-

[1]https://github.com /THUDM/cogdl

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

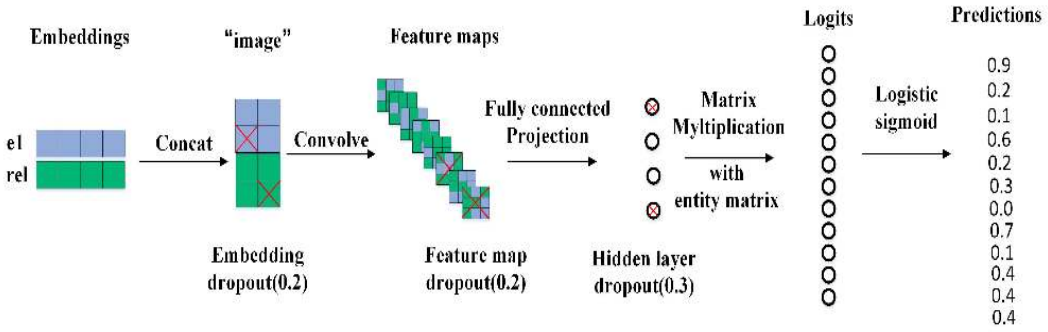

FIGURE 10: ConvE model. [94]

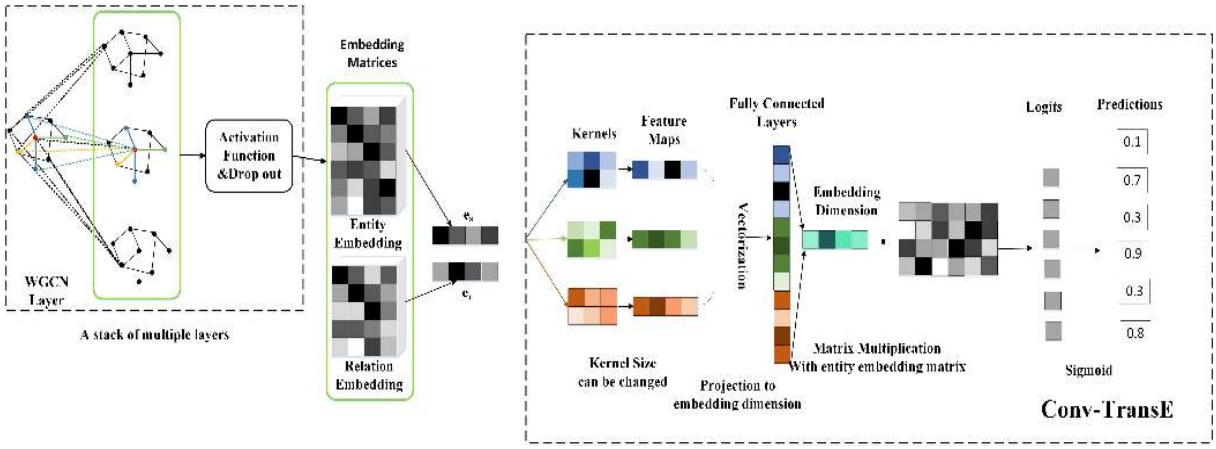

FIGURE 11: SCAN model. [98]

cient representation. But it can only manipulate the regular Euclidean data such as images (two-dimensional grid), texts (one-dimensional sequence). Graph neural network has such features as local connection, shared weights and multilayer structure [95]. These features are suitable for solving the problem of graph data. First of all, the graph is the most representative of local connection structure. Second, compared with the traditional spectrum diagram theory [96], shared weight reduces the computational cost. Finally, multilayer structure is the key to dealing with hierarchical models. It can capture different characteristics. Based on this, the convolutional neural network is extended to non-Euclidean graphs to start the research on graph neural network. Among them, graph convolutional networks (GCN) [97] are used in knowledge graph completion task much more. Another method of learning graph node embedding is obtained through the connected structure of graphs. Shang et al. combined the characteristics of GCN and ConvE model and proposed an end-to-end Structure-Aware Convolutional Network (SACN) [98] to complete the knowledge graph. SACN includes a weighted graph convolution network as a coder, aggregating relationship types of knowledge graph node structure, node attributes and edges, with learnable weights which adapts to information from adjacent structures used in local aggregation. The nodes properties in the graph are represented as additional nodes in encoder, so that they can represent graph nodes more accurately. SCAN is an improvement on the model ConvE proposed in literature [94], and it improves the performance of ConvE. In addition, ConvE does not merge the connectivity structure in the knowledge graph into representation space. The graph convolution network (GCN) used in the SCAN model is an effective tool to create the embedding representation of nodes, which can aggregate local information in its graph neighborhood for each node [99], [100], [101]. The structure of the SCAN model is shown in FIGURE 11. Graph Convolutional Network (GCN) has been widely used in the field of knowledge graph completion based on representation learning in recent years. GCN is a natural extension of convolutional neural networks to graph data. It can simultaneously perform end-to-end learning for node feature information and structure information, which is currently the best choice for graph data learning tasks [95]. Researchers have proposed a variety of improved models based on GCN, which combined the potential information of various knowledge graphs and improved the graph completion effect. In literature [102], a graph neural network-Graph convolutional network(R-GCN) for data modeling with highly multi-relational characteristics is proposed and applied to the task of completing the standard knowledge graph: link prediction, that is, completing the knowledge

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

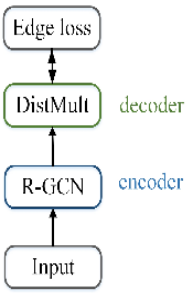

FIGURE 12: R-GCN model. [102]

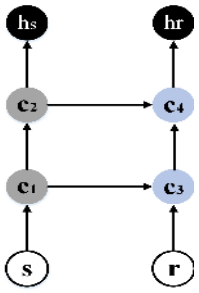

FIGURE 13: Two-layer DSKG model. [106]

graph by recovering the missing facts. The proposal of this model is to make it as an extension of GCN from local graph neighborhood to large-scale relational data. The graph completion model in this paper also adopts an automatic encoder structure, including an encoder and a decoder. The encoder uses R-GCN to complete the representation learning of the potential features for entities. And the decoder uses the tensor decomposition model DistMult [25] to predict the labeled edge according to the learned representation. The model structure is shown in FIGURE 12.

For a labeled directed graph $G = (V, E, R)$, where V, E, and R are sets of nodes, edges, and relationships respectively. The R-GCN model applies a non-linear multi-layer convolution model to graph G. Its single-layer neural network has the following Equation (10):

$$h_{(i)}^{l+1} = \sigma\left(\sum_{r\in R}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right). \tag{10}$$

This function defines an information transfer model to calculate the forward update of a single node in the relationship graph. It improves that all edges in the ordinary GCN model share the same weight W. Different types of edges (relationships) use different weights W_r and only the same edge (relationship) will use the same weight. The weights are defined as Equation (11) :

$$W_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} V_b^{(l)}. \tag{11}$$

Literature [103] further optimized the R-GCN model. If a subset of the entity set can be directly connected to the central entity, all entities in this subset have some of the same attributes, and the relationship between these entities and the central entity may be similar. These similar attributes and relationships can be abstractly aggregated into virtual entities and virtual relationships to better extract topological relationship features.

The summary of the knowledge graph completion method based on convolutional neural network is shown in Table 4.

ii)Knowledge graph completion based on sequence learning

Recurrent Neural Network (RNN) and its improved model Long Short Term Memory (LSTM) are the main models used in knowledge graph completion based on sequence learning. In the traditional neural network model, data is transmitted from the input layer to the hidden layer and then to the output layer. Since each layer is fully connected, nodes are disconnected between different layers. The purpose of this kind of neural network model is used to process sequence data. The triples in knowledge graph can be approximately regarded as a length of 3 simple sentences, such as triples (Beijing, capitalOf, China) can be converted into sentences that Beijing is the capital of China, so this kind of neural network can be used for the modeling of knowledge graph. RNN has performed well in many natural language processing (NLP) tasks, such as language modeling and machine translation [65], [66]. Literature [106] based on RNN model puts forward Deep Sequential Model for Knowledge Graph Completion (DSKG) for the knowledge graph completion, and applies a specific sampling method for model training, solving the problems that RNN will meet in the application of knowledge graph data: i) Triple group is not a natural language. Short sequences converted from triples may not provide enough contextual information for prediction. And because the number of paths in large-scale knowledge graph is huge, the cost and difficulty to build valuable long sequence is great [106]; ii) in triples, relationships and entities are two different types of elements that appear in a fixed order. Treating them as elements of the same type in a sequence will lose some characteristics. To solve these problems, DSKG uses different RNN units for the entity layer and the relationship layer. As shown in FIGURE 13, it is a double-layer DSKG model structure, in which c1, c2 and c3 and c4 are different RNN units. Entities and relationships are fed into RNN cells as sequence elements of different types and then looped over them. For an RNN unit, the previous hidden state h and sequence elements (entities and relationships, i.e. s and r) are used as inputs to predict the next hidden state h. Through experiments on knowledge graph benchmark data sets FB15K [107], WN18 [108] and FB15K [109], DSKG is better than a variety of previous models in many evaluation indicators.

The LSTM model is an improved method for RNN models encountering the problem of gradients disappearance in processing long sequence data. At each sequence index position, three gated structures are introduced: forget gate, input gate and output gate, to add or delete information. Literature [109]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

TABLE 4: Methods based on convolutional neural network

Model	Defect/Improvement
convE [94]	The connectivity structure in the knowledge graph is not merged into the representation, and the contradiction between data size and overfitting cannot be achieved
SACN [98]	Remove the steps of reconstructing for the representation of entities and relationships in ConvE, and perform convolution filtering when the entity dimensions and relationship dimensions are the same, improve the performance of ConvE
GCN [97]	Extend the convolutional neural network to non-Euclidean graph
R-GCN [102]	Modeling for multi-relational data, as an extension of GCN from local graph neighborhoods to large-scale relational data

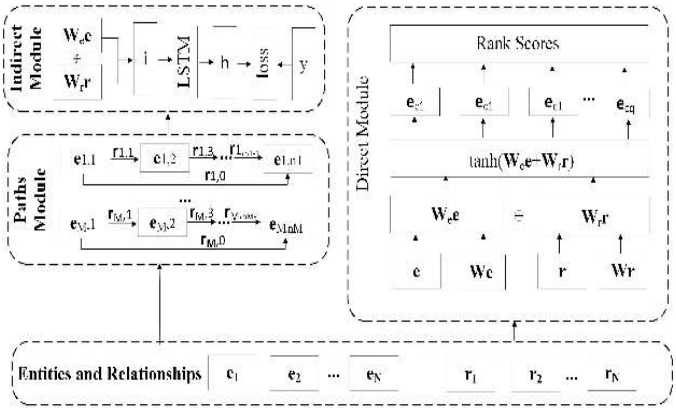

FIGURE 14: TransP model. [109]

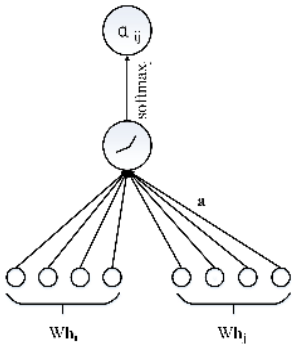

FIGURE 15: GAT layer. [112]

proposed a knowledge graph completion method based on long short-term memory network LSTM. The previously proposed translation model does not fully consider the indirect relationship between entities, which affects the accuracy of knowledge representation learning. Therefore, based on long short-term memory neural networks and existing translation models, the literature [70] proposed a multi-module hybrid neural network model TransP. The translation models represented by the Trans series are used to represent entities and relationships as embedded vectors in a low-dimensional continuous vector space, constructing an entity relationship path to describe the direct relationship between entities. LSTM is used to mine the indirect relationship between entities, which applies the optimization goal proposed in ProjE [110] to reduce the collective ranking loss, and uses level-by-level-training mechanism to optimize network parameters, so as to effectively mine the rich relationship information between entities and improve graph completion effect, as shown in FIGURE 14, **e** and **r** are the vector representations of entities and relationships, which are constantly adjusted during the training process. E and r in the indirect relationship module represent the entities and relationships included in the path constructed by the path building module; W_r and W_e are the weight matrix of relationship and entity, and e_{ci} represents all candidate entities in the direct relationship module. The probability of their establishment is respectively determined by calculation of score functions for entities and relationships.

Attention mechanisms are often used in sequence-based presentation learning methods [111]. One of the benefits of using attentional mechanisms is that variable-size inputs are allowed to be processed and then decisions are made based on the most relevant part of the input. When attentional mechanisms are used to compute the representation of a single sequence, they are often referred to as self-attentional mechanisms. The study of Vaswani, et al. [112] proved that the introduction of attention mechanism can effectively improve the method based on RNN. Based on this conclusion, the literature [112] put forward a kind of representation learning method based on attention mechanism, Graph Attention Network (GAT) as shown in FIGURE 15, whose main idea is to make neighborhood nodes involved in the implicit expression of the calculation of each center node through attention mechanism. The graph attention network has achieved good results compared with other benchmark methods on graph analysis tasks, indicating that the reference of the attention mechanism can effectively improve the expression ability of the model.

Lei et al. proposed a path reasoning method for knowledge graph completion task that integrates attention mechanism and LSTM model [113], combining the implicit semantic information representation of entities and relationships for knowledge reasoning to complete knowledge graph.

The summary of the sequence-based knowledge graph completion model is shown in Table 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030076, IEEE Access

TABLE 5: Methods based on sequence model

Model		Characteristics
Original Models	RNN [104]	Used to process data with sequence characteristics
	LSTM [109]	An improved method for the problem of gradient disappearance encountered by the RNN model in processing long sequence data
	Attention Mechanism [111]	The attention mechanism allows the processing of variable-sized inputs and makes decisions based on the most relevant part of the input.When the mechanism is used to calculate the presentation of a single sequence, it is usually called the self-attention mechanism
Improved Models	DSKG [106]	DSKG uses different RNN units for the entity layer and the relationship layer
	TransP [109]	Consider the indirect relationship between entities and improve the accuracy of knowledge representation learning
	GAT [112]	Through the self-attention mechanism,the neighbor nodes are involved in the calculation of the implicit expression of each central node

5) Application of Semantic Information in the Knowledge Graph Completion Based on Representation Learning

Semantic information can be used as additional information to assist model feature extraction to improve performance. Literature [114] considers the effect of semantic layering phenomenon in link prediction methods, that is, entities at the highest semantic level correspond to the root node of the tree, entities with lower semantic levels are closer to leaf nodes, the distance between the root node and entities at the same semantic level is equal. According to this phenomenon, the entity relationship of the knowledge graph is modeled, and the modulus and angles between the entity nodes are integrated for link prediction. Literature [115] proposed the Community-enhanced NRL (CNRL) model, the hidden community (topic) information in the knowledge graph network is embedded in the node representation, learning the community distribution of each node through Linear Discriminant Analysis (LDA) [116], and assigning the corresponding community to the nodes in the sequence label. Then on the basis of the Skip-gram model, the node representation of the central node and the corresponding community representation are used to simultaneously predict the neighboring nodes in the random walk sequence. Yang et al. proposed the Text-associated Deep Walk (TADW) model in [117]. Under the matrix factorization framework, the text features of nodes are introduced into network representation learning. The Context-aware Embedding (CANE) model [118] uses the text information of network nodes to interpret the relationship between nodes, and learns context-related network representations for network nodes according to different neighbors. Literature [119] uses ontology information to improve the effect of link classification and link prediction.

C. EVALUATION INDICATORS OF KNOWLEDGE GRAPH COMPLETION ALGORITHM

Commonly used knowledge graph completion task evaluation indicators include Hits@k, Mean Rank (MR), and Mean Reciprocal Rank (MRR). Hits@k (k is generally 10) indicates the probability of correct prediction in the top k candidate triples calculated by the algorithm. The calculation formula is shown in Equation (12):

$$H@k = \frac{\left|\{q \in Q : q < k\}\right|}{\left|Q\right|}, \tag{12}$$

in which, q represents the prediction/recommendation item, and Q represents all the prediction/recommendation items given by the algorithm. The value of Hits@k is between 0 and 1. The larger the value, the better the algorithm works.

Mean Rank is the average value of the ranking of predictions/recommendations among all candidates. The smaller the value of MR, the better the prediction effect of the model. The calculation formula is shown in Equation (13).

$$MR = \frac{1}{|Q|}\sum_{q \in Q} q. \tag{13}$$

Mean Reciprocal Rank scores the predicted triples based on whether they are true or not. If the first predicted triple is true, its score is 1, and the second true score is $\frac{1}{2}$, and so on. When the n-th triplet is established, it is scored $\frac{1}{n}$, and the final MRR value is the sum of all the scores. The calculation formula is shown in Equation (14).

$$MRR = \frac{1}{|Q|}\sum_{q \in Q} \frac{1}{q}. \tag{14}$$

The larger the MRR value, the better the model effect [120].

Hits@k represents the ability of the algorithm to correctly predict the relationship between triples, that is, the evaluation of the ability of the knowledge graph completion algorithm to predict the correct triples, and it is an indispensable evaluation index for the knowledge graph completion algorithm. The MR index can reflect the ranking of the correct triples in the probability of establishing the test triples, and is a measure of the accuracy of the knowledge graph completion algorithm. MRR is a commonly used index to measure the effect of search algorithms. If only the Top1 result of the knowledge graph is returned, the accuracy or recall will be poor, so multiple results are returned first to avoid large errors in the prediction results. The above three indicators evaluate the performance of the knowledge graph completion algorithm from different aspects. They are generally used in the evaluation of the knowledge graph completion method to

comprehensively consider the performance of the algorithm, and the algorithm of the above indicators is not complex, so it is very efficient for evaluating knowledge graph completion algorithm in different scale.

IV. PROBLEMS OF KNOWLEDGE GRAPH COMPLETION

Knowledge graph is essentially a semantic network, which is a formal description of things in the real world and their relationships [3]. For general knowledge graphs, there exist a large number of entities and concepts in the real world, which forms complex and diverse relationships between them. Correspondingly, large-scale knowledge graphs generally include tens of millions of entities and hundreds of millions of relationships. So it is an important goal of general knowledge graphs to integrate more entity relationship information as comprehensively as possible. As one of the basic technologies in the field of knowledge graphs, an important challenge facing the knowledge graph completion model is how to perform inference and calculation on large-scale knowledge graph data, that is, whether the model has the ability to handle large-scale complex relational data, including the model's ability to resolve computational efficiency and complexity conflicts and the model's completion accuracy.

Another problem faced by the knowledge graph completion task is how to design an algorithm to adapt to a dynamically changing knowledge graph . Since the real world is in constant change, so the knowledge graph should be iteratively updated, that is the new knowledge need constantly to be added into knowledge graph to extend its structure and to correctly map and describe the real world. A key challenge of it is the fusion of multi-source information, like in the more commonly used in the field of multi-language knowledge base DBpedia, YAGO, etc., a main problem is how to link and fuse information of different languages and the information in the knowledge graph correctly. The important principle of multi-source information fusion is to transform the information into a form consistent with the existing structured information of the knowledge graph during the completion process, which increases the difficulty of the knowledge graph completion. Especially for the completion based on large-scale general knowledge graph, while focusing on the quality of completion, the efficiency of the algorithm should be also considered.

As for the industry knowledge graph for a specific field, although the scope of the industry data it faces is small, the construction standard of the industry knowledge graph is different from the requirement of the breadth of income knowledge information in the open domain knowledge graph, and the industry knowledge graph is more focused on the depth of knowledge field. At the same time, there are higher requirements for the accuracy of the constructed map. Compared with better academic value of general knowledge graph, the industry knowledge graph is used much more in engineering applications and production fields, with more complicated data structure, focusing on mining limited data types, attributes and other information to complete the graph. In terms of model design, it is necessary to consider how to obtain more patterns and type information from less data to assist in the mining of graph completion knowledge. At the same time, it is also important to ensure the efficiency of the algorithm, a very important consideration in the engineering field.

V. APPLICATIONS OF KNOWLEDGE GRAPH COMPLETION

Knowledge graph completion technology has the following typical applications: 1) Knowledge graph completion is an important part of knowledge graph construction. Knowledge graph usually adopt top-down or bottom-up ways for building. For top-down building knowledge graph, the builder will determine the pattern structure and ontology at the beginning of the building, namely the conceptual form. Then knowledge instances (including entity instance example, relationship and attribute instance) are filled into constructed knowledge graph framework. At this time the applied knowledge instances mining and link is knowledge graph completion technology. At present, knowledge graph is usually constructed by combining bottom-up and top-down approaches. Ontology is constructed from bottom-up by obtaining relevant entities from massive Internet data, and then more knowledge graph instances are mined from top-down to complete the graph. The construction of knowledge graph is a dynamic process, which needs to constantly supplement the missing entities and relationships to improve the graph structure. With the help of knowledge graph completion technology, the constructed knowledge graph can be updated and extended iteratively to assist the construction of knowledge graph.

2) As knowledge graph completion is a prediction of entities, relationships, attributes and other information, and it is an important means to discover new knowledge, knowledge graph completion technology has been widely used in many upper-level tasks of knowledge graphs, such as assisting the question-answering system to achieve fast and accurate information retrieval and acquisition. Assuming that the information contained in a user's search statement or keyword is a knowledge instance with a certain correlation, we can analyze user's query sentence or keywords, extract the corresponding knowledge instance like entity, relationship, attribute, etc., to complete the missing part of existing knowledge graphs, such as supplementing missing entities and additional information in existing maps, or mining information about potential relationships between entities to improve the knowledge graph through continuous iterations and improve search engine accuracy.

Knowledge graph completion technology shows great value increasingly in the financial field, such as to forecast the financial market situation. Since financial market has strong real-time performance, namely in financial market, market information dynamically changes over time in the scene, we can implement dynamic prediction of financial market by financial knowledge graph completion based on financial events and factors [115]. The time series neural

network structures such as long and short term memory neural network will be fused with the attention mechanism to construct the prediction model. The model integrates the overall macro dynamic sequence of market and the dynamic sequence of each sub-market. The development prediction applied in financial events is used to realize the dynamic completion of financial knowledge graph. And the introduction of attention mechanism can more effectively filter out more valuable information for the current market situation. For financial market prediction scenario, the model needs to model the event dependencies of the features and the common impact of the multi-sequence inputs.

Because of the ability to organize and mine information, knowledge graph has also shown great application potential in medical diagnosis. Many researchers have devoted to the research of medical knowledge graph. At present, most medical institutions have constructed electronic medical records based on the medical records of patients, and summarized a large amount of electronic medical record information, including basic patient information, medical history, symptoms, etiology and corresponding treatment methods, and drug information, extracting entities and relationships and attributes to build medical knowledge graph. When using the constructed medical knowledge graph to assist intelligent diagnosis of diseases, the basic information of the patient can be linked with the existing medical map, and then the knowledge graph completion technology is applied to mine the disease entity information related to the patient entity to link together, so as to effectively provide more accurate reference information for the patient's disease diagnosis.

3) From general aspect, knowledge graph is a kind of graph data, and knowledge graph completion is to mine the implicit relationship between vertices in the graph or predict new vertices and edges that have a potential relationship with the existing graph structure through the graph structure of existing data (including points and edges). From the perspective, knowledge graph completion model and algorithm can also be applied to all kinds of graph structure prediction task scenarios based on graph data, such as all kinds of application scenarios like social and financial risk control graph database, etc.

VI. SUMMARY AND PROSPECTS

This article analyzes and summarizes the existing mainstream methods of knowledge graph completion, and divides them into traditional knowledge graph completion methods and deep learning-based knowledge graph completion methods from the perspective of evolution. The former is mainly related to knowledge inferences based on rules or graphs. Although it can achieve good graph completion effects, it is difficult to use on large-scale knowledge graphs due to the high computational complexity. In recent years, representation learning methods represented by deep learning have received widespread attention. By efficiently calculating the semantic connections between entities and relationships in low-dimensional spaces, they effectively solve the problem of data sparseness and make the overall accuracy and efficiency significantly improved.

Based on the analysis of existing research methods, the study is expected to continue in the following directions.

1) Knowledge graph completion of multi-class information fusion. The most important difference between knowledge graph and other graph data is that knowledge graph is a kind of semantic network. If reasoning is only based on its graph network structure, a lot of useful information will be lost and the expected completion effect cannot be achieved. Considering the semantic information hidden in the semantic network, including the types and attributes of entities and relationships, semantic connections between entities, domain rule information and other multi-source information, more comprehensive feature could be extracted to improve the learning ability of the KGC model, and thus achieve more accurate prediction and completion.

2) Knowledge Graph Completion based on graph neural network model. Knowledge graph completion based on graph neural networks applies powerful learning and expression ability to model knowledge graphs, which can acquire good reasoning ability [27]. Firstly, in the aspect of knowledge representation, more features such as semantic information can be considered to build the graph vector representation for boosting the machine learning. Secondly, a key point is to construct a suitable graph neural network model, and use its powerful expression and reasoning capabilities to achieve a more accurate and comprehensive knowledge graph completion.

3) Expansion of knowledge graph completion in upper application fields. Knowledge graph completion technology has direct applications in Internet related fields such as intelligent question answering and search engine. In recent years, knowledge graph completion has also been widely applied as an auxiliary development technology in the fields of finance and medical treatment. At present, one of the development directions of knowledge graph completion is how to expand the breakthrough of this technology in more application scenarios to help high efficiency output in engineering application and production field.

4) How to improve the knowledge graph completion technology to assist the realization of dynamic knowledge graph. Although the construction of knowledge graphs is in an iterative updating process, the existing knowledge graphs are still relatively static. Such knowledge graphs can no longer meet the real-time requirements of people for dynamic information, and most application fields rely on the timeliness of information. Dynamic knowledge graph has been a major trend in the current development, and its realization is inseparable from the graph inference completion technology.

5) The ability of knowledge graph completion to mine new knowledge plays an important role in the development process of turning knowledge graph to cognitive graph. Cognitive graph can be understood as a dynamic system integrating knowledge reasoning ability. Although there is still controversy about whether the knowledge graph will

eventually move towards the cognitive graph, from the perspective of the human cognitive process, the static knowledge graph containing inherent knowledge will never be the end of development. If the dynamic knowledge graph is used as the basis to combine with the cognitive ability of human beings, the real artificial intelligence can be expected.

ACKNOWLEDGMENT

This work was supported in part by the Funds for Key Research and Development Plan Project of the Shaanxi Province, China, under Grant 2017GY-072, 2019ZDLGY17-08, 2020ZDLGY09-02.

REFERENCES

[1] J. Li and L. Hou, "Review of knowledge graph research," *Journal of Shanxi University: Natural Science Edition*, vol. 40, no. 3, pp. 454-459, Mar. 2017.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008*, Vancouver, BC, Canada, pp. 1247-1250, Jun. 2008.

[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: a nucleus for a web of open data," *The Semantic Web*, Berlin, Heidelberg, pp. 722-735, 2007.

[4] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yagočža core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web - WWW '07*, Banff, Alberta, Canada, pp. 697-706, May. 2007.

[5] J. Ding and W. Jia, "The research advances of knowledge graph completion algorithm," *Information and Communications Technologies*, no. 1, pp. 56-62, Apr. 2018.

[6] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embedding for knowledge graph completion," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence - AAAI '15*, Austin, Texas, pp. 2181-2187, Jan. 2015.

[7] S. C. Shapiro, "A net structure for semantic information storage, deduction, and retrieval," in *Proceedings of the 2nd International Joint Conference on Artificial Intelligence*, Los Altos, CA, pp. 512ÍC523, Sep. 1971.

[8] G. Qi, H. Gao, and T. Wu, "The reaserch advances of knowledge graph," *Technology Intelligence Engineering*, vol. 3, no. 1, pp. 004-025, 2017.

[9] A. Singhal, "Introducing the Knowledge Graph: things, not strings, " *Google Blog*, https://www.blog.google/products/search/introducing-knowledge-graph-things-not/, 2012.

[10] M. Zhu, B. Bao, and C. Xu, "Research progress on development and construction of knowledge graph," *Journal of Nanjing University of Information Science and Technology*, vol. 9, no. 6, pp. 575-582, Jun. 2017.

[11] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the Association for Information Science and Technology,*, vol. 58, no. 7, pp. 1019-1031, 2007.

[12] H. Xiao, M. Huang, and X. Zhu, "From one point to a manifold: knowledge graph embedding for precise link prediction," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence - IJCAI '16*, Arizona State University, pp. 1315-1321, Jul. 2016.

[13] T. Trouillon, J. Welbl, S. Riedel, Gonzé. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - ICML '16* vol. 48, pp. 2071-2080, Jun. 2016.

[14] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* Lisbon, Portugal, pp. 705-714, Sep. 2015.

[15] R. Xie,Z. Liu and M. Sun, "Representation learning of knowledge graphs with hierarchical types," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence - IJCAI '16*, Arizona State University, pp. 2965-2971, Jul. 2016.

[16] D. Nguyen, K. Sirts, L. Qiu, and M. Johnson, "Neighborhood mixture model for knowledge base completion," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany, pp. 40-50, Agu. 2016.

[17] T. Jiang, T. Liu, T. Ge, L. Sha, S. Li, B. Chang, and Z Sui, "Encoding temporal information for time-aware link prediction," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, USA, pp. 2350-2354, 2016.

[18] Y. Lin, Z. Liu, and M. Sun, "Knowledge representation learning with entities, attributes and relations," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence - IJCAI '16*, Arizona State University, pp. 2866-2872, Jul. 2016.

[19] B. Shi and T. Weninger, "Open-world knowledge graph completion," *arXiv preprint arXiv:1711.03438*, Nov. 2017.

[20] R. Reiter, "On closed world data bases(Logic and Data Bases)", *Springer, MA*, Boston, pp. 55-76, Jan. 1977.

[21] S. Wang, Z. Du, and X. Meng, "Research progress of large-scale knowledge graph completion technology", *Scientia Sinica Informationis*, vol. 50, no. 4, pp. 551-575, Apr. 2020.

[22] J. Liang, S. Zhang, and Y. Xiao, "How to keep a knowledge base synchronized with its encyclopedia source," in *Proceedings of the Twenty-Sixth Inter- national Joint Conference on Artificial Intelligence - IJCAI qŕ17*, Melbourne, Australia, pp. 3749-3755, Aug. 2017.

[23] H. Shah, J. Villmow, A. Ulges, U. Shwanecke, and S. Faisal, "An open-world extension to knowledge graph completion models," *arXiv preprint arXiv:1906.08382*, pp. 3044-3051, 2019.

[24] Q. Chen, K. Chen, S. Wu, L. Shou, and G. Chen, "Research about knowledge graph completion based on active learning", *Journal of Frontiers of Computer Science and Technology*, vol. 14, no. 5, pp. 769-782, 2020.

[25] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutiérrez, J. E. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A. N. Ngomo, S. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, "Knowledge Graphs," *arXiv preprint arXiv:2003.02320*, Apr. 2020.

[26] S. Guan, X. Jin, Y. Jia, Y. Wang, and X. Cheng, "Knowledge reasoning over knowledge graph: a survey," *Journal of Software*, vol. 29, no. 10, pp. 2966-2994, 2018.

[27] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, USA, pp. 1306-1313, 2010.

[28] Y. Wei, J. Luo, and H. Xie, "KGRL: an OWL2 RL reasoning system for large scale knowledge graph," in *12th International Conference on Semantics, Knowledge and Grids (SKG)*, Beijing, China, pp. 83-89, 2016.

[29] W. Y. Wang, K. Mazaitis, W. W. Cohen, "Programming with personalized pagerank: a locally groundable first-order probabilistic logic," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, New York, America, pp. 2129-2138, May. 2013.

[30] W. Y. Wang, K. Mazaitis, N Lao, T, Mitchell, and W. W. Cohen, "Efficient inference and learning in a large knowledge base: Reasoning with extracted information using a locally groundable first-order probabilistic logic," *Machine Learning*, vol. 100, no. 1, pp. 101-126, Apr. 2014.

[31] C. Fu, T. Chen, M. Qu, W. Jin, and X. Ren, "Collaborative policy learning for open knowledge graph reasoning," *arXiv preprint arXiv:1909.00230*, Aug. 2019.

[32] H. Paulheim and C. Bizer, "Improving the quality of linked data using statistical distributions," *International Journal on Semantic Web and Information Systems*, vol. 10, no. 2, pp. 63-86, 2014.

[33] H. Zhang, L. Wang, and Y. Chen, "Research progress of probabilistic graphical models: a survey," *Journal of Software*, vol. 11, pp. 2476-2497, 2013.

[34] S. Zhai, L. Guo, S. Gao, H. Duan, Z. Li, and Y. Ma, "A method for knowledge graph completion based on bayesian reasoning," *Journal of Chinese Computer Systems*, vol. 39, no. 5, pp. 995-999, Jul. 2018.

[35] D. George, W. Lehrach, K. Kansky, M. Lazaro-Gredilla, C. Laan, B. Marthi, X. Lou, Z. Meng, Y. Liu, H. Wang, A. Lavin, and D. Phoenix, "A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs," *Science*, vol. 358, no. 6368, pp. 2612-2621, Oct. 2017.

[36] F. Kou, J. Du, C. Yang, Y. Shi, M. Liang, Z. Xue, and H. Li, "A multi-feature probabilistic graphical model for social network semantic search," *Neurocomputing*, vol. 336, no. 7, pp. 67-78, Apr. 2019.

[37] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107-136, 2006.

[38] S. Jiang, D. Lowd, and D. Dou, "Learning to refine an automatically extracted knowledge base using markov logic," in *2012 IEEE 12th International Conference on Data Mining*, Brussels, pp. 912-917, 2012.

[39] D. Heckerman, G. Dan, and D. M. Chickering, "Learning Bayesian networks: the combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197-243, 1995.

[40] C. N. Sekharan, "A probabilistic graphical model for learning as search," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference*, Las Vegas, USA, pp. 1-4, Jan. 2017.

[41] O. Pourret, P. Naim, and B. Marcot, "Introduction to Bayesian networks," *John Wiley and Sons*, New Jersey, America, pp. 1-13, 2008.

[42] L. Han, Z. Yin, Y. Wang, K. Hu, and K. yue, "Link prediction of knowledge graph based on Bayesian network," *Journal of Frontiers of Computer Science and Technology*, vol. 11, no. 5, pp. 742-751, 2017.

[43] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Machine Learning*, vol. 81, no. 1, pp. 53-67, Oct. 2010.

[44] N. Lao, T. Mitchell, and W. W. Cohen, "Random walk inference and learning in a large scale knowledge base," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK, pp. 529-539, Jul. 2011.

[45] Q. Wang, J. Liu, Y. Luo, and B. Wang, "Knowledge base completion via coupled path ranking," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pp. 1308-1318, 2016.

[46] E. Miller, "An introduction to the resource description framework," *Bulletin of the American Society for Information Science and Technology*, vol. 25, no. 1, pp. 15-19, 1998.

[47] Z. Liu, M. Sun, Y. Lin, and R. Xie, "Knowledge representation learning: a review," *Journal of Computer Research and Development*, vol. 53, no. 2, pp. 247-261, 2016.

[48] M. Nickel, V. Tresp, and H. Kriegel, "A Three-Way Model for Collective Learning on Multi-Relational Data," in *Proceedings of the 28th International Conference on International Conference on Machine Learning - ICML '11*, Gramado, Brazil, vol. 11, pp. 809-816, Jun. 2011.

[49] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, and K. Murphy "Knowledge vault: a web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, USA, pp. 601-610, 2014.

[50] K. Hayashi and M. Shimbo, "On the equivalence of holographic and complex embeddings for link prediction," *arXiv preprint arXiv:1702.05563*, 2017.

[51] R. Jenatton, N. Roux, A. Bordes, and G. Obozinski, "A latent factor model for highly multi-relational data," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Siem Reap, Cambodia, vol 2, pp. 3167-3175, Dec. 2012.

[52] X. Glorot, A. Bordes, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233-259, 2014.

[53] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, pp. 1105-1114, Aug. 2016.

[54] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[55] A. Bordes, N. Usunier, A. G. Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *27th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, NV(US), vol. 4, pp. 2799-2807, Dec. 2013.

[56] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the Twenty-Eighth AAAI conference on artificial intelligence - AAAI '14*, North America, pp. 1112-1119, Jun. 2014.

[57] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China, vol. 1, pp. 687-696, 2015.

[58] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, pp. 985-991, Feb. 2016.

[59] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39-41, 1995.

[60] S. Ma, J. Ding, W. Jia, K. Wang, and M. Guo, "TransT: type-based multiple embedding representations for knowledge graph completion," *Machine Learning and Knowledge Discovery in Databases*, vol. 10534, pp. 717-733, Jan. 2017.

[61] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embedding," *arXiv preprint arXiv:1904.10281*, 2019.

[62] Z. Sun, Z. Deng, J. Nie, and J. Tang, "RotatE: knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[63] H. Xiao, M. Huang, Y. Hao, and X. Zhu, "TransA: an adaptive approach for knowledge graph embedding," *arXiv preprint arXiv:1509.05490*, 2015.

[64] H. Xiao, M. Huang, and X. Zhu, "TransG: a generative model for knowledge graph embedding," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, vol. 1, pp. 2316-2325, Jan. 2016.

[65] M. Fan, Q. Zhou, E. Chang, and F. Zheng, "Transition-based knowledge graph embedding with relational mapping properties," in *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, Phuket, Thailand, pp. 328-337, Dec. 2014.

[66] Y. Fang, X. Zhao, Z. Tan, S. Yang, and W. Xiao, "A revised translation-based method for knowledge graph representation," *Journal of Computer Research and Development*, vol. 55, no. 1, pp. 139-150, 2018.

[67] B. Yang, W. Yih, X. He, J Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *3rd International Conference on Learning Representations(ICLR 2015)*, San Diego, CA, USA, May. 2015.

[68] M. Nickel, L. Rosasco, and T. Poggio, "Holographic Embeddings of Knowledge Graphs," in *3Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, pp. 1955-1961, 2016.

[69] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding:a survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724-2743, Dec. 2017.

[70] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, vol. 70, pp. 2168-2178, Aug. 2017.

[71] X. Zhou, Q. Zhu, P. Liu, and L. Guo, "Learning knowledge embeddings by combining limit-based scoring loss," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, pp.1009-1018, 2017.

[72] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *26th Advances in Neural Information Processing Systems, NIPS 2013*, Harrahs and Harveys, Lake Tahoe, USA, pp. 926-934, Dec. 2013.

[73] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 932-938, Jan, 2000.

[74] A. Paccanaro and G. E. Hinton, "Learning distributed representations of concepts from relational data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, pp. 200-212, 2000.

[75] A. Paccanaro and G. E. Hinton, "Learning distributed representations of concepts using linear relational embedding," *Neural Nets WIRN Vietri-01*, Toronto, Canada, pp. 134-143, 2002.

[76] A. Bordes, J. Weston, R. Collobert, Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proceedings of the 25th Conference on Artificial Intelligence*, San Francisco, USA, pp. 301-306, Aug. 2011.

[77] F. Giunchiglia, M. Yatskevich, P. Shvaiko, "Semantic matching: algorithms and implementation," *Journal on Data Semantics*, vol. 9, pp. 540-512, 2007.

[78] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, vol. 10, pp. 1511-1517, 2017.

[79] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining,ACM* Francisco, California, USA, pp. 855-864, Aug. 2016.

[80] B. Perozzi, R. Al-Rfou and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, New York, USA, pp. 701-710, Aug. 2014.

[81] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web, International World Wide Web*

Conferences Steering Committee, Florence, Italy, pp. 1067-1077, May. 2015.

[82] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *JJournal of machine learning research*, vol. 12, no. 8, pp. 2493-2537, 2011.

[83] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modeling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[84] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[85] D. Wang, C. Peng, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, pp. 1225-1234, 2016.

[86] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, pp. 1105-1114, 2016.

[87] J. Qiu, X.Yu, H. Ma, J. Li, K. Wang, and J Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459-467. 2018.

[88] J. Qiu, X.Yu, H. Ma, J. Li, C. Wang, K. Wang, and J. Tang, "Netsmf: Large-scale network embedding as sparse matrix factorization," in *World Wide Web Conference*, San Francisco, United States, pp. 1509-1520. 2019.

[89] J. Zhang, Y. Dong, Y. Wang, and J. Tang, "ProNE: fast and scalable network representation learning," in *Proceedings of the 28th Int. Joint Conference on Artificial Intelligence*, Macau, China, pp. 4278-4284. 2019.

[90] C. Tu, C. Yang, Z. Liu, M. Sun, "Network representation learning: an overview," *SCIENTIA SINICA Informationis*, vol. 47, no. 8, pp. 980-996, 2017.

[91] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *32st AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, Feb. 2018.

[92] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," in *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11-33, 2015.

[93] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: Algorithms, applications and open challenges," in *International Conference on Computational Social Networks*, Evanston, Illinois, USA, pp. 79-91, Jul. 13th - 15th, 2018.

[94] F. R. K. Chung and F. C. Graham, "Spectral graph theory," *American Mathematical*, Soc., 1997.

[95] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[96] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-End structure-aware convolutional networks for knowledge base completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA, pp. 3060-3067, Jan. 27th - Feb. 1th, 2019.

[97] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *30th Advances in Neural Information Processing Systems, NIPS 2017*, Long Beach Convention Center, Long Beach, USA, pp. 1024-1034, Dec. 4th - 10th, 2017.

[98] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[99] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1611.07308*, 2016.

[100] M. Schlichtkrull, T. N. Kipf, F. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*, Heraklion, Crete, Greece, pp. 593-607, June . 3th - 7th, 2018.

[101] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint arXiv:1602.02410*, 2016.

[102] A. Tian, C. Zhang, M. Rang, X. Yang, and Z. Zhan, "RA-GCN: Relational aggregation graph convolutional network for knowledge graph completion," in *Proceedings of the 12th International Conference on Machine Learning and Computing*, HShen Zhen, China, pp. 580-586, Feb. 2020.

[103] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, G. Hinton, "Grammar as a foreign language," in *28th Advances in Neural Information Processing Systems, NIPS 2015*, Montreal, Canada, pp. 2773-2781, Dec. 7th - 9th, 2015.

[104] L. Guo, Q. Zhang, W. Ge, W. Hu, and Y. Qu, "DSKG: A Deep Sequential Model for Knowledge Graph Completion," in *China Conference on Knowledge Graph and Semantic Computing*, Tianjin, China, pp. 65-77, Aug. 15th - 18th, 2018.

[105] K. Toutanova, and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, Beijing, China, pp. 57-66, Jul., 2015.

[106] P. Zeng, Q. Tan, X. Meng, H. Zhang, and J. Xu, "Modeling complex relationship paths for knowledge graph completion," *IEICE TRANSACTIONS on Information and Systems*, vol. 101, no. 5, pp. 1393-1400, 2018.

[107] B. Shi, and T. Weninger, "ProjE: Embedding projection for knowledge graph completion," in *31st AAAI Conference on Artificial Intelligence*, San Francisco, California USA, pp. 112-122, Feb. 4th - 9th, 2017.

[108] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[109] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," *arXiv preprint arXiv:1611.02344*, 2016.

[110] R. Fu, Z. Zhang, and L. Li, "Attention is all you need," in *30th Advances in Neural Information Processing Systems, NIPS 2017*, Long Beach Convention Center, Long Beach, USA, pp. 5998-6008, Dec. 4th - 10th, 2017.

[111] K. Lei, J. Zhang, Y. Xie,D. Wen, D. Chen, and M. Yang, "Path-based reasoning with constrained type attention for knowledge graph completion," *Neural Computing and Applications*, vol. 35, no. 189, pp. 1-10, 2019.

[112] H. Zhao, L. Wu, Z. Li, X. Zhang, Q. Liu and E. Chen, "Predicting the dynamics in Internet finance based on deep neural network structure," *Journal of Computer Research and Development*, vol. 56, no. 8, pp. 1621-1631, 2019.

[113] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," *arXiv preprint arXiv:1911.09419*, 2020.

[114] C. Tu, H. Wang, X. Zeng, Z. Liu, and M. Sun, "Community-enhanced network representation learning for network analysis," *arXiv preprint arXiv:1611.06645*, 2016.

[115] T. Griffiths and M. Steyvers, "Finding scientific topics," in *Proceedings of the National Academy of Sciences of the United States of America*, pp. 5228ÍC5235, 2004.

[116] C. Yang, Z. Liu, D. Zhao, and M. Sun, "Network representation learning with rich text information," in *Proceedings of the 24th International Conference on Artificial Intelligence*, Buenos Aires, pp. 2111ÍC2117, Jul. 2015.

[117] C. Tu, H. Liu, Z. Liu, and M. Sun, "CANE: context-aware network embedding for relation modeling," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouve, pp. 1722ÍC1731, 2017.

[118] E. Amador-DomÍnguez, P. Hohenecker, T. Lukasiewicz, D. Manrique, E. Serrano, "An ontology-based deep learning approach for knowledge graph completion with fresh entities," in *Proceedings of the 16th International Conference on Distributed Computing and Artificial Intelligence*, Berlin, Heidelberg, vol. 1003, 2019.

[119] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, and C. Li, "Realistic re-evaluation of knowledge graph completion methods: An experimental study," *arXiv preprint arXiv:2003.08001*, Mar. 2020.

[120] Z. Xu, Y. Sheng, L. He, and Y. Wang, "Review on knowledge graph techniques," *Journal of University of Electronic Science and Technology of China*, vol. 45, no. 4, pp. 589-606, Jul. 2016.

ZHE CHEN, born in 1969. Ph.D, Associate Professor. He is a member of CCF. His research interests include machine learning and image processing.

YUEHAN WANG, born in 1995. Master candidate. Her research interests include data mining and knowledge graph.

BIN ZHAO, born in 1996. Master candidate. His research interests include urban computing, machine learning and artificial neural networks.

JING CHENG, born in 1997. Master candidate. Her research interests include data mining and knowledge graph.

XIN ZHAO, born in 1999. He is an undergraduate of Chang'an University. He is interested in image processing and deep learning.

ZONGTAO DUAN, born in 1977. Ph.D, Professor. He is a member of CCF, CCF High Performance Computing, and Pervasive Computing Technical Committee. His research interests include information processing and application.

Dr. Duan is a member of CCF, CCF High Performance Computing, and Pervasive Computing Technical Committee.