# A Survey on Translating Embedding based Entity Alignment in Knowledge Graphs

Jin Jiang*, Mohan Li* and Zhaoquan Gu*
*Cyberspace Institute of Advance Technology*
*Guangzhou University*
Guangzhou, China
*Email:2112006019@e.gzhu.edu.cn, {limohan, zqgu}@gzhu.edu.cn

***Abstract*—Knowledge Graph (KG) as an ideal knowledge base can effectively support the mining, analysis and reasoning of complex relational data. It has been widely used by academia and industry. Entity alignment (EA) is one of the basic tasks of KG fusion. Its main goal is to align heterogeneous entities that refer to the same but from different sources. In recent years, a lot of researches have focused on this task. This paper presents a systematic survey of the KG EA based translating embeddings. The purpose is to provide a complete and systematic overview of these methods and challenges. Furthermore, we discuss the future research trends and the correlation with MDATA. Our detailed review can offer technical assistance for researchers or engineers who want to quickly have a comprehensive understanding about the KG EA and the trend lines.**

***Index Terms*—entity alignment, knowledge graph, translating embeddings**

## I. Introduction

Knowledge graph, which was proposed by Google in 2012 [1], is essentially a semantic network that consists of the nodes and edges in directed graph. Initially, KG is structured to improve web search engine on account of it can availably capture the adjacent knowledge of a node in the structural knowledge representation graph. In just a few years, KG has been widely used in many fields of computer science, such as search engines [2], big data analytics [3,4] and recommender system [5]. There are already many open source KGs, such as DBpedia [6], Freebase [7] and YAGO [8].

The purpose of building up KGs is to empower AI techniques to learn much more knowledge to boost the cognitive abilities of AI systems. However, it's extremely hard to achieve. One of the core issues is that the richness of KG will determine the effectiveness of KG-based tasks. In order to obtain a KG that contains more knowledge, one of the common ideas is to merge multiple KGs from different sources into a larger KG. EA is an effective technology to expand the knowledge graph by fusing with different KGs. Figure 1 shows a simple example of EA on two KGs of diverse knowledge bases. The same entities in different KGs are linked by using EA techniques. For example, calculating the similarity of each entities or reasoning the equivalent to each other. Unfortunately, due to the heterogeneous representation for KGs, comparing with the similarity of different KGs is often not highly effective. These technologies often rely on special designs, which limits their scope of application.

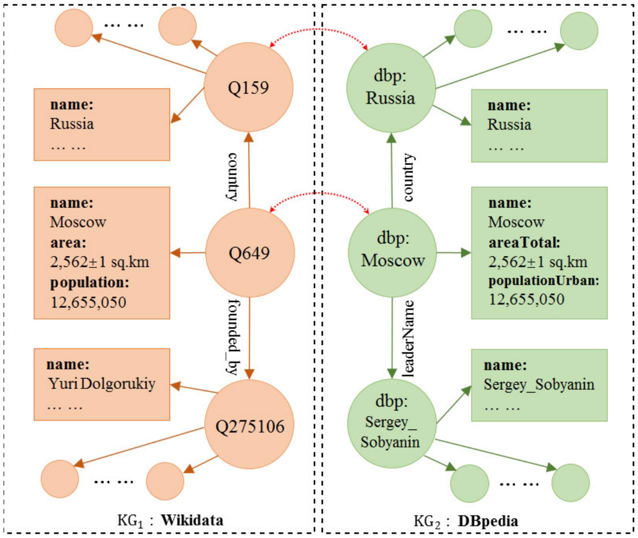

Fig. 1. A simple example of EA. (Red dashed lines indicate the aligned entities.)

Since 2013, when the word2vec [9,10] was proposed, knowledge embedding based on deep learning grabs the academy's attention for its ability to capture the potential semantic information, which improves the effect of EA. [11,12] Current strategies of EA mainly contain two methods, GCN-based [13] and translation-based [14]. Given an actual KG, the methods of GCN-based can aggregate the information of a node and its neighbors in the graph to generate an embedding. The methods of translation-based is essentially learning the structure information of nodes in the graph. Both of the two methods can learn embeddings of the nodes, so it is meaningfully to compare the distance of the embeddings in vector space to find the aligned entities. In cross-KGs EA task, most of the techniques utilize pre-existing aligned entities as label data to unsupervised or semi-supervised learning. Compared to traditional techniques for KG EA, the techniques of embedding are more suitable to EA task. However, the embedding representations pay more attention on structure information and need a lot of labeled data for learning, it

978-1-6654-1815-7/21/$31.00 ©2021 IEEE
DOI 10.1109/DSC53577.2021.00033

also does not work perfectly for EA tasks. Recently, many of works focus on learning the embeddings which joint structure information and attribute information. How to effectively extract and use these information is a major challenge in the KG EA task.

In this paper, we focalize the light on the KG EA task based translating embeddings. The rest of the survey is organized as follows. Section II provides preliminaries, including the definition of KG, problem formulation and traditional EA techniques. As a core part of the paper, Section III surveys the most representative translation-based EA techniques. Section IV introduces some common KG datasets. Section V gives a summary and discusses future directions.

## II. Preliminaries

### A. Knowledge Graph (KG)

A KG is a directed graph composed of nodes and edges representing entities/attributes and relations/predicates, respectively. It can be formally represented by $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{A}, \mathcal{V}, \mathcal{T})$, where $\mathcal{E}, \mathcal{R}, \mathcal{A}$ and $\mathcal{V}$ denote the sets of entities, relation predicates, attribute predicates and attribute values, respectively. $\mathcal{T} = \mathcal{T}_r \cup \mathcal{T}_a$ denotes knowledge in the form of a set of triples, where $\mathcal{T}_r$ is a relation triple, $\mathcal{T}_a$ is a attribute triple. There are two types of triples, relation triples in the form of $(h, r, t) \in \mathcal{T}_r$ and attribute triples in the form of $(e, a, v) \in \mathcal{T}_a$ . A triple $(h, r, t)$ indicates a relation predicate $r$ between a head entity $h$ and a tail entity $t$, where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. For example, `(China, Capital, Beijing)` means the head entity `China` has relation of `Capital` with the hail entity `Beijing`. A triple $(e, a, v)$ indicates an attribute predicate $a$ between an entity $e$ and attribute values $a$, where $e \in \mathcal{E}$ , $a \in \mathcal{A}$ and $v \in \mathcal{V}$. For example, `(China, Area, "960 million sq.km")` means the attribute values `"960 million sq.km"` is an attribute `Area` of the entity `China`.

### B. Entity Alignment (EA) Task

Given two different KGs $\mathcal{G}^{(1)} = \left(\mathcal{E}^{(1)}, \mathcal{R}^{(1)}, \mathcal{A}^{(1)}, \mathcal{V}^{(1)}, \mathcal{T}^{(1)}\right)$ and $\mathcal{G}^{(2)} = \left(\mathcal{E}^{(2)}, \mathcal{R}^{(2)}, \mathcal{A}^{(2)}, \mathcal{V}^{(2)}, \mathcal{T}^{(2)}\right)$ , the EA task aims to find and unify entities with the same actual meaning in the two KGs. That means to identify every pair of entities $\left(e^{(1)}, e^{(2)}\right)$ in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, where $e^{(1)} \in \mathcal{E}^{(1)}$, $e^{(2)} \in \mathcal{E}^{(2)}$ and $e^{(1)} \equiv e^{(2)}$.

The EA task is to identify the same real-world entity pairs, which completed by calculating the similarity of the embedding representations between a source entity and a target entity. In Figure 2, we give a generic framework for translation-based EA techniques as illustrated. The embedding representations of entities in the KG, which learned by knowledge representation (KR) module, are encoded by the EA module based on the translation technique. Existing EA models often require seed alignments $S = \left\{\left(e_i^{(1)}, e_j^{(2)}\right) | e_i^{(1)} \in \mathcal{E}^{(1)}, e_j^{(2)} \in \mathcal{E}^{(2)}; e_i^{(1)} \equiv e_j^{(2)}\right\}$ as label data, where " $\equiv$" denotes the equivalence relationship and it means the $i$-th entity $e_i^{(1)} \in \mathcal{E}^{(1)}$ equivalent to the $j$-th entity $e_j^{(2)} \in \mathcal{E}^{(2)}$. We can solve the EA task through training the triples $T_s$ in the KG, where $T_s = \{T_i^{(1)}, T_j^{(2)}\}$ denotes the triples corresponding to the sets of seed alignments.

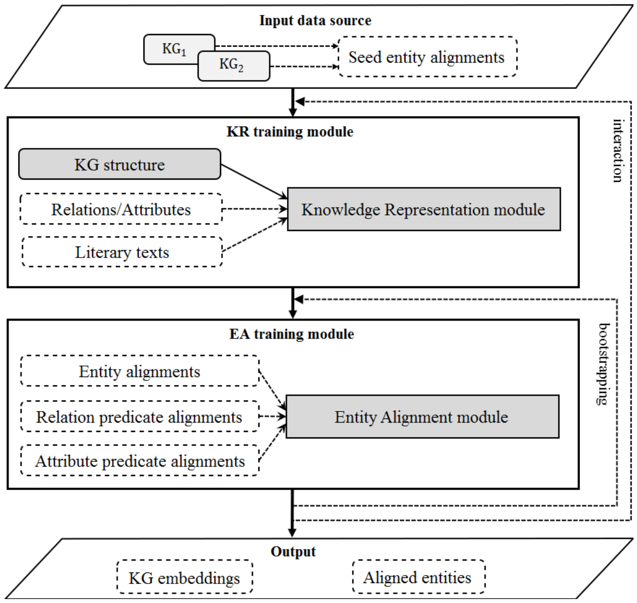

Fig. 2. Framework of Translation-based EA techniques. (Dotted lines indicate optional parts.)

Due to the KGs from different sources often have a mass of noise and incompleteness, EA task faces the following major challenges.

- **Entity disambiguation**: The knowledge representation in diverse KGs may be totally different. For example, the symbolic representation of KGs in different languages may be completely different though they are exactly the same concept.
- **Insufficient EA seeds**: Most of the solutions rely on the seed alignments, but the labeled seed data requires a huge amount of manual work and is difficult to obtain.
- **Missing features**: In the research of EA, a variety of methods are often used to capture the features with various perspectives. However, it is complicated to fully utilize these information in the EA task whether extracting structural features or attribute features.
- **Poor robustness**: If there are incorrect triples in the KGs or get some wrong seed alignments during the training process, it will lead to error propagation and difficult to revise.

For replying these challenges, a number of EA techniques are proposed in recent years. The core conception is fully utilize the structure information in graph and the semantic information of entities to improve the accuracy of EA task. We will details these techniques in the upcoming section.

### C. Traditional EA Techniques

Most of traditional techniques for EA use hand-crafted features [15] and crowdsourcing [16], all of these need to

TABLE I
THE UNIVERSAL SYMBOLS

| Symbols | Descriptions |
|---|---|
| $\mathcal{G}$ | A knowledge graph which consists of $\mathcal{E}$, $\mathcal{R}$, $\mathcal{A}$, $\mathcal{V}$ and $\mathcal{T}$ |
| $\mathcal{E}$ | A set of entities $(h, t, e \ \in \mathcal{E})$ |
| $\mathcal{R}$ | A set of relation predicates $(r \in \mathcal{R})$ |
| $\mathcal{A}$ | A set of attribute predicates $(a \in \mathcal{A})$ |
| $\mathcal{V}$ | A set of attribute values $(v \in \mathcal{V})$ |
| $\mathcal{T}$ | A set of triples, which may consists of relation triples $\mathcal{T}_r$ and attribute triples $\mathcal{T}_a$ ($\mathcal{T} = \mathcal{T}_r \cup \mathcal{T}_a$ or $\mathcal{T} = \mathcal{T}_r$) |
| $(h, r, t)$ | A relation triple in $\mathcal{T}_r$ , which consist of a head entity $h$, a tail entity $t$, a relation predicate $r$. The corresponding embeddings is $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ |
| $(e, a, v)$ | A attribute triple in $\mathcal{T}_a$ , which consist of an entity $e$, an attribute value $v$, an attribute predicate$a$. The corresponding embeddings is $(\mathbf{e}, \mathbf{a}, \mathbf{v})$ |
| $S$ | A set of pre-aligned entities from $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. $S = \left\{ \left( e_i^{(1)}, e_j^{(2)} \right) \mid e_i^{(1)} \in \mathcal{E}^{(1)}, e_j^{(2)} \in \mathcal{E}^{(2)}; e_i^{(1)} \equiv e_j^{(2)} \right\}$ |
| $T_s$ | A set of triples for the each entity in $S$, $T_s = \{T_i^{(1)}, T_j^{(2)}\}$ |

be well-designed by a domain expert. Earlier EA techniques use string similarity as the main method. For example, SILK [17] proposed the Link Specification Language to allow user-defined similarity measures for comparing the specific type of attributes. RDF-AI [18] provides an alignment framework, which using fuzzy string matching based on word relation [19], sequence alignment [20] and taxonomic similarity. LIMES [21] computes an approximation of entity similarity by using the triangle inequality.

These traditional EA techniques, highly rely on rules to identify similar entities, are hard to satisfy the intricate and vast EA tasks. Obviously, the accuracy and generality of these techniques are extremely weak.

## III. TRANSLATION-BASED ENTITY ALIGNMENT

In this section, we will review in detail the EA techniques based on the translation model. The universal symbols in our notation convention are summarized in Table 1. Nearly all EA techniques based translating embeddings are implemented on the baseline model TransE [14]. TransE is an extraction technique of relational structure information in the KG. Similar entities can be found by comparing the structure information of entities in different KGs, then the EA task can be completed. However, in the process of generating embedding, TransE only focus on the relational structure of KGs, which lost an amount of attribute information. Therefore, in order to improve the effect of EA, recent researches often focus on extracting both the structure information and attribute information. Later in the chapter, we will explain the TransE model in detail first, then discuss the EA techniques which only use structure information for KG embedding. Finally, we will discuss the EA techniques which exploit structure and attribute information.

### A. The TransE model

TransE is a knowledge representation model that is proposed by Bordes et al. in 2013 [14]. The key idea of the model is expressing the relation between the KG entities as translation operations in a vector space. The idea of the translation model originated from the phenomenon of semantic translation between word vectors, which is discovered in the word2vec by Mikolov et al. in 2013. As an example, $v\,(\texttt{China}) - v\,(\texttt{Beijing}) \simeq v\,(\texttt{Japan}) - v\,(\texttt{Tokyo})$, where $v\,(\cdot)$ denotes the word vector representation process. The phenomenon shows that the implicit semantic relation information between words is effectively encoded into word vectors. Inspired by this discovery, TransE take the triple $(h, r, t)$ in KG as vector representation $(\mathbf{h}, \mathbf{r}, \mathbf{t})$, which is called embeddings. More importantly, TransE regards the relation $\mathbf{r}$ as a translation vector from a head entity $\mathbf{h}$ to the corresponding tail entity $\mathbf{t}$, then $h + r \cong t$. The score function is defined as follows:

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

Ideally, the value of the $f_r$ is zero. To learn the embeddings, TransE will attempt to minimize a margin-loss function blow:

$$\mathcal{L}_{\text{transe}} = \sum_{(h,r,t) \in \mathcal{T}_r} \sum_{\left(h^{'}, r, t^{'}\right) \in \mathcal{T}_r^{'}} \max\left(0, \left[\gamma + f_r\,(h, r, t) - f_r\left(h^{'}, r, t^{'}\right)\right]\right) \tag{2}$$

where $\mathcal{T}_r$ is a set of correct triples, $\mathcal{T}_r^{'}$ is a set of corrupted triples that generated from correct triples with randomly replacing either the head or the tail entity. Here, $T_r^{'} = \{(h', r, t) | h' \in E\} \cup \{(h, r, t') | t' \in E\}$ . And $\gamma > 0$ is a hyper-parameter which be used to expand the gap between the score of correct triples and corrupted triples.

Although TransE is pretty efficient in one-to-one relations and irreflexive, it also exists some logically flaws when encounters many-to-one, one-to-many and many-to-many relations. To deal with the weakness of TransE, there also have some variants of the TransE, such as TransH [22], TransR [23], TransD [24] and PTransE [25].

### B. EA Techniques Only Exploit Structure Information

The original EA techniques only exploit structure information by training translation model. In this case, the method always contains a KR module and a EA module. The KR module inherits the pros and cons of TransE, so it only works well for the simple relation KG. The EA module primarily use a linear transformation matrix to align different KG. The matrix is trained by seed alignments, which means we should gain sufficient label data to complete the project.

*a) MTransE:* MTransE is proposed by Chen et al. in 2016 [26], which is the first EA model based on the TransE. MtransE can be divided into two modules, including knowledge representation (KR) module and entity alignment (EA) module. The KR module directly use TransE model to generate embeddings. For two KGs $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, different vector spaces are obtained through independent training by minimizing the loss function $\mathcal{L}_{\text{transe}}$ respectively. Then, the EA module relies on the seed alignments to learn a linear transformation matrix between different KGs. The alignment score function is defined as:

$$\begin{aligned} & f_{\text{mtranse}}\left(T_i^{(1)}, T_j^{(2)}\right) = \\ & \left\|M_{ij}^e h_i^{(1)} - h_j^{(2)}\right\| + \left\|M_{ij}^r r_i^{(1)} - r_j^{(2)}\right\| + \left\|M_{ij}^e t_i^{(1)} - t_j^{(2)}\right\| \end{aligned} \tag{3}$$

Where the $M_{ij}^e$ and $M_{ij}^r$ are the linear transformation matrix of entity and relation embeddings respectively. They shifts the KG structure embedding of $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ into a same vector space by minimizing the following loss function:

$$\mathcal{L}_a = \sum_{\left(T_i^{(1)}, T_j^{(2)}\right) \in T_s} f_{\text{mtranse}}\left(T_i^{(1)}, T_j^{(2)}\right) \tag{4}$$

During the training of the MTransE, the distance of embeddings between the seed alignments in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ is moving closer and closer in the same vector space. The overall loss function is using a hyperparameter $\alpha$ to balance the KR module and EA module as follow:

$$\mathcal{L}_{\text{mtranse}} = \mathcal{L}_{\text{transe}} + \alpha \mathcal{L}_a \tag{5}$$

The method of MTransE relies on the quantity and quality of seed alignments. However, KGs from diverse sources are often heterogeneous and full of noise, and it is difficult to obtain sufficient seed alignments. In addition, MTransE is a rational model, which assumes that the embeddings of two independently constructed KGs can be merged simply be a linear transformation. This hard transformation actually loses quite a number of useful information. Therefore, subsequent researchers have conducted more explorations on the problems of limited seeds and missed features.

*b) IPTransE:* IPTransE is proposed by Zhu et al. in 2017 [27]. Aiming at the problem that seed alignments is difficult to obtain, an iterative EA module that only needs a small number of seed alignments is designed. IPTransE consists of three module: KR module, joint embeddings module and iterative EA module. The KR module can adopt TransE or PTransE, PTransE model connects entities with relation paths instead of single relation in KG. The KR module treats each KG separately but not considers the correspondence of entities between different KGs. So joint embeddings module designs the following three strategies to employs the seed alignments.

- Translation-based strategy

Similar to the idea of translation method, assuming that there is a relation $r^{(\mathcal{E}_1 \to \mathcal{E}_2)}$ between the embeddings of the entities $e_i^{(1)} \in \mathcal{E}^{(1)}, e_j^{(2)} \in \mathcal{E}^{(2)}$ in the seed alignments $S$ such that $e_i^{(1)} + r^{\left(\mathcal{E}^{(1)} \to \mathcal{E}^{(2)}\right)} \simeq e_j^{(2)}$. The energy function is defined as:

$$E\left(e_i^{(1)}, e_j^{(2)}\right) = \left\|e_i^{(1)} + r^{\left(\mathcal{E}^{(1)} \to \mathcal{E}^{(2)}\right)} - e_j^{(2)}\right\| \tag{6}$$

- Linear transformation strategy

Similar to the idea of MTransE method, learning a transformation matrix $M^{\left(\mathcal{E}^{(1)} \to \mathcal{E}^{(2)}\right)}$ to closer the distance between seed alignments in the vector space such that $M^{\left(\mathcal{E}^{(1)} \to \mathcal{E}^{(2)}\right)} e_i^{(1)} \simeq e_j^{(2)}$. The energy function is defined as:

$$E\left(e_i^{(1)}, e_j^{(2)}\right) = \left\|M^{\left(\mathcal{E}^{(1)} \to \mathcal{E}^{(2)}\right)} e_i^{(1)} - e_j^{(2)}\right\| \tag{7}$$

- Parameter sharing strategy

Assuming that the same entities between seed alignments have an overlapping relationship, so that the aligned entities share the same embeddings in the two KGs. Due to the $e_i^{(1)} \equiv e_j^{(2)}$, $\left(e_i^{(1)}, e_j^{(2)}\right) \in S$, the parameter sharing strategy forces $e_i^{(1)} = e_j^{(2)}$.

After joint embedding, new aligned entities can be captured, which can be used to compose more seed alignments. Hence the iterative EA module attempts to identify new seed alignments by discovering entities with high similarity in the process of embedding two KGs into a unified space, and iteratively labels the new seed alignments as training data. In the process of adding the seed alignments, IPTransE designs a hard EA strategy and a soft EA strategy. Compared with directly adding the aligned entities to share the embeddings, the soft EA strategy evaluates the reliability of each new aligned entities, which can reduce the impact of error propagation.

*c) BootEA:* BootEA is proposed by Sun et al. in 2018 [28], which is an EA model based on bootstrapping from both labeled data and unlabeled data. In order to solve the problem of lacking sufficient seed alignments $S$, BootEA iteratively labels the newly caught aligned entities as label training data by identifying entities with high similarity, when embedding the two KGs into an unified vector space. The label data will be used to learn the alignment-oriented embeddings of KGs.

The KR module of BootEA not only uses the loss function of TransE to learn all the correct triples in the source KGs, but also utilizes recombined triples by bootstrapping strategy. These recombined triples generates by replacing an entity to another in the seed alignments. With the iteration of bootstrapping, $S$ grows gradually. The loss function of the embedding is same as TransE, formulated as $\mathcal{L}_e$ below:

$$\begin{aligned} \mathcal{L}_e = & \sum_{(h,r,t) \in \mathcal{T}_r} \max\left(0, \left[f_r\left(h, r, t\right) - \gamma_1\right]\right) + \\ & \lambda \sum_{\left(h', r, t'\right) \in \mathcal{T}_r'} \max\left(0, \left[\gamma_2 - f_r\left(h^{'}, r, t^{'}\right)\right]\right) \end{aligned} \tag{8}$$

Here, the hyperparameter $\gamma_1$ and $\gamma_2$ are used to minimize the $f_r\left(h, r, t\right)$ and $f_r(h^{'}, r, t^{'})$ respectively. $\lambda$ is a negotiable parameter to balance the influence of corrupted triples on the embeddings.

For another, the EA module is treated as a classification task. BootEA thinks the corresponding entity of an entity as a label, and iteratively learns a classifier by self-expanding from labeled and unlabeled data. All the entities in $S$ are computed by the following cross-entropy loss function:

$$\mathcal{L}_a = -\sum_{e_1 \in \mathcal{E}_1} \sum_{e_2 \in \mathcal{E}_2} \varphi_{e_1}\left(e_2\right) \log \pi\left(e_2 | e_1\right) \tag{9}$$

Where $\pi$ is a classifier that predicts the aligned entity $e_2$ from a given entity $e_1$. $\varphi_{e_1}\left(e_2\right)$ is the probability that $e_2$ has equivalence relationship for $e_1$. If $e_2$ is the true alignment entity for $e_1$, $\varphi_{e_1}\left(e_2\right) = 1$, therefore the probability distribution $\varphi_{e_1}$ of $e_1$ will fully concentrate on $e_2$. Otherwise, $\varphi_{e_1}$ will be a uniform distribution, if $\varphi_{e_1}(e_2) = 0$.

Combining the KR module and the EA module can get the overall objective function $\mathcal{L}_{\text{bootea}} = \mathcal{L}_e + \mathcal{L}_a$

*d) SEA:* SEA is proposed by Pei et al. in 2019 [29], which is a semi-supervised training model based on Generative Adversarial Networks (GAN) [30]. Researchers had observed that the frequency of each entity in the KG has a remarkable impact on the EA task. For example, in the EA task of English-French KGs, if the word `"Washington"` exists more frequent appearance in English KG than French KG therefore the embeddings of the word are not close to each other in vector space.

To solve this problem, inspired by GAN, SEA uses adversarial training to reduce the impact from entity frequency and also takes the loss function of TransE in the KR module. Specifically, the module designs a discriminator $D_1$ to distinguish between high-frequency and intermediate-frequency entities, and also designs a discriminator $D_2$ to distinguish between intermediate-frequency and low-frequency entities. Here presumed the loss function of $D_1$ and $D_2$ are $\mathcal{L}_{D_1}$ and $\mathcal{L}_{D_2}$ respectively. Therefore, the objective function of the KR module is as $\mathcal{L}_e = \mathcal{L}_{\text{transe}} - \mu\mathcal{L}_{D_1} - \mu\mathcal{L}_{D_2}$, where $\mu$ is a balance parameter.

The EA module of SEA, resembling the MTransE, demands linear transformation matrices to accomplish the mapping of the vector space. And considering the limit of seed alignments, the EA module designs a semi-supervised training model to optimize the generation of transformation matrix. In this case, there exists a permutation matrix $M^{(1)}$ that satisfies $M^{(1)}e^{(1)} \equiv e^{(2)}$ for the label data $\left(e^{(1)}, e^{(2)}\right)$ in the $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. And by the same logic, existing a permutation matrix $M^{(2)}$ that satisfies $M^{(2)}e^{(2)} \equiv e^{(1)}$. Apart from that, considering that the transformation matrix should satisfy cyclic consistency, such as $e^{(1)} \equiv M^{(2)}M^{(1)}e^{(1)}$ and $e^{(2)} \equiv M^{(1)}M^{(2)}e^{(2)}$. Hence both labeled entities and unlabeled entities fulfill the condition of cyclic consistency, all the entities are handled into the following loss function:

$$\begin{aligned}\mathcal{L}_a = \alpha_1 & \sum_{\left(e^{(1)}, e^{(2)}\right) \in S} \left\| M^{(1)}e^{(1)} - e^{(2)} \right\| + \left\| M^{(2)}e^{(2)} - e^{(1)} \right\| \\ & + \left\| M^{(2)}M^{(1)}e^{(1)} - e^{(1)} \right\| + \left\| M^{(1)}M^{(2)}e^{(2)} - e^{(2)} \right\| \\ + \alpha_2 & \sum_{\left(e^{(1)}, e^{(2)}\right) \in \mathcal{T} - S} \left\| M^{(2)}M^{(1)}e^{(1)} - e^{(1)} \right\| + \left\| M^{(1)}M^{(2)}e^{(2)} - e^{(2)} \right\|\end{aligned} \tag{10}$$

Here, the weight parameter $\alpha_1$ and $\alpha_2$ are used to balance the labeled data and unlabeled data.

### *C. EA Techniques that Exploit Structure and Attribute Information*

The translation-based models can only have an ideal representation effect for triples with one-to-one relation, it is not effective for complex relation triples. A mass of usable information is lost on account of only using the structure information. In order to make full use of the semantic features of the triples in the KG, researchers have begun to mine the attribute information of the entity itself. Through observation, it is found that there exists plentiful attribute triples in KG. For example, the triple `(China, Area, "960 million sq.km")` expresses the area attribute of the country. Therefore, the triples in KGs can be divided into relation triples $(h, r, t)$ and attribute triples $(e, a, v)$, which are treated differently in the progress of training.

*a) JAPE:* JAPE is proposed by Sun et al. in 2017 [31], which is a cross-language EA joint attribute embeddings model. JAPE learns the structure information of two KGs into a unified vector space, and then refines it by leveraging the attribute information.

JAPE is also composed of a KR module and an EA module, but the KR module includes two parts, structure embedding and attribute embedding. During learning the structure embeddings, the KR module uses TransE to train $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ respectively, meanwhile generates corresponding structure embeddings $E_s^{(1)}$ and $E_s^{(2)}$. While learning the attribute embeddings, the module uses the Skip-Gram [8] algorithm to generate word embeddings for each data type of attribute values, and treats the resulting word embeddings as attribute embeddings. Then an attribute-based entity embedding matrix $E_a^{(i)}$ $(i = 1, 2)$ can be formed, which is composed of the averaged attribute embeddings of all entities.

After obtaining $E_s^{(i)}$ and $E_a^{(i)}$, JAPE calculates the cross-KG similarity $S^{(1),(2)}$ according to the attribute-based entity embedding matrices. Similarly, we can calculate the inner-KG similarity $S^{(1)}$ and $S^{(2)}$:

$$\begin{aligned} & S^{(1),(2)} = E_a^{(1)} E_a^{(2)^\top}; \\ & S^{(1)} = E_a^{(1)} E_a^{(1)^\top}; S^{(2)} = E_a^{(2)} E_a^{(2)^\top} \end{aligned} \tag{11}$$

*b) KDCoE:* KDCoE is proposed by Chen et al. in 2018 [32], which adjusts the structure embeddings through the text description of the entities. KDCoE consists of multilingual entity embedding module and multilingual description embedding module. The entity embedding module is same as MTransE. Then the description embedding module uses a encoder, which composed of two stacked attention GRU layers, to model the description of the two languages' view.

Furthermore, we can use structure information and entity descriptions to perform cross-language reasoning.

*c) AttrE:* AttrE is proposed by Zhang et al. in 2019 [33]. It has an unusual property that the model can achieve excellent effects though it not requires the pre-seed alignments. AttrE is composed of a predicate alignment module, a KR module and an EA module. In the predicate alignment module, the relation in a triple is regarded as a predicate, and the matching is performed according to the literal similarity of predicate. For example, if we finds two highly similar predicates `population` in $\mathcal{G}^{(1)}$ and a predicate `populationTotal` in $\mathcal{G}^{(2)}$, then renames them based a unified naming scheme (e.g., `population`). After completing the predicate alignment, we can divide all triples into relation triples $(h, r, t)$ and attribute triples $(e, a, v)$ according to whether the object after the predicate is an entity or an attribute value. In the KR module, the structure embeddings are learning based TransE.

On the other hand, the attribute embeddings are generated by using a combination function $f_a\left(a\right)$ to encode the attribute values. The learning loss function of attribute embedding, which resembles in TransE, is formulated as:

$$\mathcal{L}_a = \sum_{a \in \mathcal{T}_a} \sum_{a' \in \mathcal{T}_a'} \max\left(0, \gamma + \alpha\left(f_s\left(a\right) - f_s\left(a^{'}\right)\right)\right) \quad (12)$$

Where $\gamma$ and $\alpha$ are hyperparameters and $f_s\left(a\right) = \|h + r - f_a\left(a\right)\|$. $f_a\left(a\right)$ can choose in direct summation, LSTM and N-gram.

The next step is joint structure embeddings and attribute embeddings. In the EA module, it can infer the head entities in two triples are identically if they have the same predicate and object. Furthermore, if it is found that the tail entity in the same relation is exactly the tail entity just aligned in another pair of triples, we can infer that the head entities of the two triples are corresponding with each other.

In fact, AttrE captures the aligned entities by using the attribute embeddings although it not has pre-seed alignments. And then more potential aligned entities are found by computing the cosine similarity between all the entities.

*d) MultiKE:* MultiKE is proposed by Zhang et al. in 2019 [34], which combines the learning embeddings from multiple views. MultiKE includes three representative views: name view, relation view and attribute view. In essence, it is a synthesis of multiple methods, expecting to obtain more features of the entity to assist in the EA task.

The name view uses Skip-Gram to generate embeddings $H^{(1)}$ for every entity in KGs. In relation view, still using the TrsanE to learn the structure embeddings $H^{(2)}$, but utilizing a sigmoid function change the value of score function to a probability value and minimizing the logistic loss function as follows:

$$\mathcal{L}_r = \sum_{(h,r,t) \in \mathcal{T}_r \cup \mathcal{T}_r'} \log\left(1 + \exp\left(\zeta\left(h,r,t\right) f_r\left(h,r,t\right)\right)\right) \quad (13)$$

Here $\mathcal{T}_r$, $\mathcal{T}_r^{'}$ are correct triples (positive examples) and corrupted triples (negative examples) respectively. $\zeta\left(h,r,t\right) = 1$, if $(h,r,t)$ is positve; otherwise $\zeta\left(h,r,t\right) = -1$.

In attribute view, MultiKE composes the attribute predicates and attribute values into attribute-value matrix $[a|v]$. Drawing on the extensively used model CNN [35] in the field of image processing, MultiKE extracts the feature vectors from attribute-value matrix and makes the entity $e$ closed to the feature vectors. The attribute embeddings $H^{(3)}$ generated by using the score function as follows:

$$f_{\text{attr}}(e,a,v) \;=\; - \;\; \|e - CNN\left([a|v]\right)\| \quad (14)$$

After gaining the three kinds of embeddings, the weighted average of multi-view embeddings are served as ultimate combination embeddings $\tilde{H}$ of entities. Finally, MultiKE maximizes the consistency between the combined embeddings and the view-specific embeddings, and try to preserve the features extracted from each view by using the follow loss function:

$$\mathcal{L} = \sum_{i=1}^{3} \left\| \tilde{H} - H^{(i)} \right\|_F^2 \quad (15)$$

Where $\|\cdot\|_F^2$ is a function for calculating the Frobenius norm.

The EA module is based on consistency inference, which means the effect of $e$ is equivalent to $\hat{e}$ if they are a same pair of entities. Supposing the probability extrapolation will never be changed when change the $e$ to $\hat{e}$ in a candidate alignment triple ( $\left\{(e,\hat{e}) \middle| e \in \mathcal{E}^{(1)}, \hat{e} \in \mathcal{E}^{(2)}\right\}$ ). For the reasoning of the identity of relation and attribute, MultiKE adopts a soft alignment algorithm which consider the similarity of names and semantics comprehensively. This algorithm facilitates the discovery of more potential alignments.

*e) COTSAE:* COTSAE is proposed by Yang et al. in 2020 [36]. The previous models weigh all attribute information without assessing whether they're effective or ineffective. However, an entity may have multiple attributes, and different attributes may have different contribution to the EA task. In response to this problem, COTSAE proposes a joint attention mechanism method to learn the importance of different attribute values to every entity.

On the one hand, the KR module uses TransE to learn structure information. On the other hand, it also uses a Pseudo-Siamese network [37] to learn the attribute information. The loss function of Pseudo-Siamese network is a contrastive loss to make the distance of the aligned entities closer in the vector space. It is worth mentioning that the attribute predicate and attribute values of an entity share the same attention weight, the $i$-th weight value is computed by the following expression:

$$\alpha_i = softmax\left(A^{\top} W_p \mathbf{a}_i\right) \quad (16)$$

Where $A$ is an embedding matrix for all attributes of an entity, $W_p$ is a learning weight matrix for an attribute predicate, $\mathbf{a}_i$ is the attribute predicate embedding of the $i$-th attribute. The weight values are shared by attribute predicate embedding $\mathbf{e}_p$ and attribute values embedding $\mathbf{e}_v$. The final embedding of the attribute predicate/value of an entity is obtained by the weighted sum of all attribute predicate/value embeddings of the entity.

COTSAE also uses the bootstrapping strategy, each iteration generates structure embeddings and attribute embeddings by TransE and Pseudo-Siamese network respectively. Furthermore, using these embeddings calculate an entity similarity matrix between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. For the entity pairs whose similarity is higher than a given threshold, constructing a bipartite graph. In each iteration, the aligned entity pairs are predicted by the inference module through graph matching on the bipartite graph, meanwhile adds them to the training set of the next iteration.

## IV. Open Knowledge Graphs

Current knowledge bases are growing and grabbing more and more abundant. In addition, the increasing amount of KGs

TABLE II
OPEN KGS

| Name | Founder | Size | Data source | Dataset formats | Query language |
|---|---|---|---|---|---|
| WordNet [40] | The Cognitive Science Laboratory of Princeton University(1985) | 207,016 word-sense pairs | Created by experts | XML, JSON-LD, RDF | WordNet-API |
| OpenCyc [41] | Cycorp Company(2001) | 1.6M triples | Created by experts | RDF, Proprietary File Format | CycQL |
| Freebase [7] | Metaweb Technologies(2007) | 1.9B triples | Wikipedia, NNDB and others | RDF, OWL | MQL |
| Dbpedia [6] | The Berlin Free University and the University of Leipzig(2007) | 900M triples | Wikipedia | RDF | SPARQL |
| YAGO [8,42] | Max Planck Institute in Germany(2007) | more than 50M entities and 2B facts | Wikipedia, WordNet, Geonames | RDF, TSV | SPARQL, Wikibase-API |
| WikiData [43] | Wikimedia Foundation(2012) | 94M items, 12. 5B triples | Created by users | JSON, XML, RDF | SPARQL, Wikidata Query |
| CN-Dbpedia [44,45] | The Knowledge Workshop Laboratory of Fudan University(2015) | 67M triples | Wikipedia, Baidu Baike | CSV | CN-DBpedia API |
| OpenKG [46] | Chinese Information Processing Society of China(2015) | 107M triples | Zhishi.me, CN-Dbpedia, and others | RDF, OWL, JSON | OpenKG-API |

have shown growth potential in areas like question answering system [38] and machine comprehension [39]. Here we will explain briefly some representative open KGs in Table 2.

## V. CONCLUSIONS AND FUTURE DIRECTIONS

In summary, the current embedding-based entity alignment task mainly revolves around two major problems: the KR task (how to embed two KGs from different sources into a unified vector space) and the EA task (how to achieve entity alignment in the unified vector space). Most of models above are still based on TransE to generate structure embeddings, and then optimize the embeddings by extracting more features from the triples, so that the final embedding representations are conducive to the EA task.

In the EA task, the spatial distance between entity embeddings is used to measure the similarity. Most of the EA methods, MTransE as an example, use seed entity alignment to mine the mapping rules of aligned entities between different KGs. However, this method relies on the quality and quantity of the pre-seed alignments. For the problem of insufficient seeds, models such as BootEA use a bootstrapping method to iteratively discover more potential aligned entities during the embedding training process. Meanwhile, in order to reduce the error propagation that caused by errors in the bootstrapping process, models such as AttrE use a soft alignment method to ensure the quality of the new seeds alignment entity.

In the foreseeable future, the EA task will still address the problem for the features loss after TransE training by using the attribute information of the entity to optimize the embeddings. For example, techniques such as GCN and CNN in the field of computer vision (CV) can be used to capture the feature information of entities. Or using the text mining techniques in the field of natural language processing (NLP) to obtain the attribute features from the literal description. These techniques all aim to dig out the richer semantic information to improve the EA effect.

Changing the view to the initial knowledge representation, it is also possible to optimize the entity alignment task through improving the origin defects of TransE itself. For example, entity in the PTransE model can be represented by multiple relationship paths, which is more flexible than TransE. Optimizing the KR model itself is more likely to solve the problem from the root. However, this is extremely difficult, not only to balance the complexity of the algorithm, but also to consider the complex and changeable relationship between knowledge. Under the framework of MDATA [47], it can assist entity alignment tasks in the dimensionality of time and space. The spatiotemporal characteristics can be used as the attributes of the entity to optimize the embedding representations. It is also possible to map entities to different hyperplanes in the vector space [48], so as to classify entities based on the spatiotemporal characteristics. The architecture of MDATA is more effective for the scenarios that the differences between entities can be distinguished mainly by time and space. However, in more complex world knowledge, how to perform effective knowledge representation to assist entity alignment tasks is still a problem.

## ACKNOWLEDGMENT

This work is funded by Guangdong Basic and Applied Basic Research Foundation (No.2021A1515012307, 2020A1515010450), Guangzhou Basic and Applied Basic Research Foundation (No. 202102021207, 202102020867), the National Natural Science Foundation of China (No. 62072130, 61702220, 61702223), Guangdong Province Key Area R&D Program of China (No. 2019B010136003, 2019B010137004), Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019), and Guangdong Higher Education Innovation Group (No. 2020KCXTD007) and Guangzhou Higher Education Innovation Group (No. 202032854).

# References

[1] A. Singhal. ”Introducing the Knowledge Graph: things, not strings“, Official Google Blog, May 2012. http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html.

[2] Steiner, Thomas, et al. “Adding realtime coverage to the google knowledge graph.“ 11th International Semantic Web Conference (ISWC 2012). Citeseer, 2012.

[3] Elluri, Lavanya, Ankur Nagar, and Karuna Pande Joshi. “An integrated knowledge graph to automate gdpr and pci dss compliance.“ 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.

[4] Z. Gu, L. Wang, X. Chen, Y. Tang, X. Wang, X. Du, M. Guizani, Z. Tian, “Epidemic Risk Assessment by A Novel Communication Station Based Method.“ IEEE Transactions on Network Science and Engineering. 2021.

[5] Zhang, Fuzheng, et al. “Collaborative knowledge base embedding for recommender systems.“ Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.

[6] Auer, Sören, et al. “Dbpedia: A nucleus for a web of open data.“ The semantic web. Springer, Berlin, Heidelberg, 2007. 722-735.

[7] Bollacker, Kurt, et al. ”Freebase: a collaboratively created graph database for structuring human knowledge.” Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008.

[8] Hoffart, Johannes, et al. “YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia.“ Artificial Intelligence 194 (2013): 28-61.

[9] Mikolov, Tomas, et al. “Efficient estimation of word representations in vector space.“ arXiv preprint arXiv:1301.3781 (2013).

[10] Mikolov, Tomas, et al. “Distributed representations of words and phrases and their compositionality.“ arXiv preprint arXiv:1310.4546 (2013).

[11] W. Han, Z. Tian, C. Zhu, Z. Huang, Y. Jia, and M. Guizani. “A Topic Representation Model for Online Social Networks Based on Hybrid Human-Artificial Intelligence.“ IEEE Transactions on Computational Social Systems. vol. 8, no. 1, pp. 191-200, Feb. 2021. DOI: 10.1109/TCSS.2019.2959826.

[12] W. Han, Z. Tian, Z. Huang, S. Li, and Y. Jia. “Topic Representation Model Based on Microblog Behavior Analysis.“ World Wide Web Journal. 23, pages3083–3097 (2020). DOI: 10.1007/s11280-020-00822-x.

[13] Kipf, Thomas N., and Max Welling. “Semi-supervised classification with graph convolutional networks.“ arXiv preprint arXiv:1609.02907 (2016).

[14] Bordes, Antoine, et al. “Translating embeddings for modeling multi-relational data.“ Neural Information Processing Systems (NIPS). 2013.

[15] Suchanek, Fabian M., Serge Abiteboul, and Pierre Senellart. “Paris: Probabilistic alignment of relations, instances, and schema.“ arXiv preprint arXiv:1111.7164 (2011).

[16] Rinser, Daniel, Dustin Lange, and Felix Naumann. “Cross-lingual entity matching and infobox alignment in Wikipedia.“ Information Systems 38.6 (2013): 887-907.

[17] Volz, Julius, et al. “Discovering and maintaining links on the web of data.“ International Semantic Web Conference. Springer, Berlin, Heidelberg, 2009.

[18] Scharffe, François, Yanbin Liu, and Chuguang Zhou. “Rdf-ai: an architecture for rdf datasets matching, fusion and interlink.“ Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US). 2009.

[19] Miller, George A. WordNet: An electronic lexical database. MIT press, 1998.

[20] Rivas, Elena, and Sean R. Eddy. “A dynamic programming algorithm for RNA structure prediction including pseudoknots.“ Journal of molecular biology 285.5 (1999): 2053-2068.

[21] Ngomo, Axel-Cyrille Ngonga, and Sören Auer. “Limes-a time-efficient approach for large-scale link discovery on the web of data.“ integration 15.3 (2011).

[22] Wang, Zhen, et al. “Knowledge graph embedding by translating on hyperplanes.“ Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 28. No. 1. 2014.

[23] Lin, Yankai, et al. “Learning entity and relation embeddings for knowledge graph completion.“ Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 29. No. 1. 2015.

[24] Ji, Guoliang, et al. “Knowledge graph embedding via dynamic mapping matrix.“ Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers). 2015.

[25] Lin, Yankai, et al. “Modeling relation paths for representation learning of knowledge bases.“ arXiv preprint arXiv:1506.00379 (2015).

[26] Chen, Muhao, et al. “Multilingual knowledge graph embeddings for cross-lingual knowledge alignment.“ *arXiv preprint arXiv:1611.03954* (2016).

[27] Zhu, Hao, et al. “Iterative Entity Alignment via Joint Knowledge Embeddings.“ *IJCAI*. Vol. 17. 2017.

[28] Sun, Zequn, et al. “Bootstrapping Entity Alignment with Knowledge Graph Embedding.“ *IJCAI*. Vol. 18. 2018.

[29] Pei, Shichao, et al. “Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference.“ *The World Wide Web Conference*. 2019.

[30] Goodfellow, Ian J., et al. “Generative adversarial networks.“ arXiv preprint arXiv:1406.2661 (2014).

[31] Sun, Zequn, Wei Hu, and Chengkai Li. “Cross-lingual entity alignment via joint attribute-preserving embedding.“ *International Semantic Web Conference*. Springer, Cham, 2017.

[32] Chen, Muhao, et al. “Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment.“ *arXiv preprint arXiv:1806.06478* (2018).

[33] Trisedya, Bayu Distiawan, Jianzhong Qi, and Rui Zhang. “Entity alignment between knowledge graphs using attribute embeddings.“ *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. No. 01. 2019.

[34] Zhang, Qingheng, et al. “Multi-view knowledge graph embedding for entity alignment.“ *arXiv preprint arXiv:1906.02390* (2019).

[35] Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. “A convolutional neural network for modelling sentences.“ arXiv preprint arXiv:1404.2188 (2014).

[36] Yang, Kai, et al. “COTSAE: CO-Training of Structure and Attribute Embeddings for Entity Alignment.“ *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. No. 03. 2020.

[37] Bromley, Jane, et al. "Signature verification using a“ siamese“ time delay neural network.“ Advances in neural information processing systems 6 (1993): 737-744.

[38] Xu, Lin, et al. “End-to-end knowledge-routed relational dialogue system for automatic diagnosis.“ Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. No. 01. 2019.

[39] Qiu, Delai, et al. “Machine reading comprehension using structural knowledge graph-aware network.“ Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

[40] Miller, George A. “WordNet: a lexical database for English.“ Communications of the ACM 38.11 (1995): 39-41.

[41] Färber, Michael, et al. “Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago.“ Semantic Web 9.1 (2018): 77-129.

[42] Rebele, Thomas, et al. “YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames.“ International semantic web conference. Springer, Cham, 2016.

[43] Vrandečić, Denny. “Wikidata: A new platform for collaborative data collection.“ Proceedings of the 21st international conference on world wide web. 2012.

[44] Xu, Bo, et al. “CN-DBpedia: A never-ending Chinese knowledge extraction system.“ International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, Cham, 2017.

[45] Xu, Bo, et al. “CN-DBpedia2: An Extraction and Verification Framework for Enriching Chinese Encyclopedia Knowledge Base.“ Data Intelligence 1.3 (2019): 271-288.

[46] Chen, Huajun, et al. “OpenKG chain: A blockchain infrastructure for Open Knowledge Graphs.“ Data Intelligence 3.2 (2021): 205-227.

[47] Jia, Yan, Zhaoquan Gu, and Aiping Li, eds. MDATA: A New Knowledge Representation Model: Theory, Methods and Applications. Vol. 12647. Springer Nature, 2021.

[48] Dasgupta, Shib Sankar, Swayambhu Nath Ray, and Partha Talukdar. “Hyte: Hyperplane-based temporally aware knowledge graph embedding.“ Proceedings of the 2018 conference on empirical methods in natural language processing. 2018.