A Survey on Knowledge Graphs: Representation, Acquisition and Applications

Shaoxiong Ji, Shirui Pan, Erik Cambria, *Senior Member, IEEE,*
Pekka Marttinen, Philip S. Yu, *Life Fellow, IEEE,*

***Abstract*—Human knowledge provides a formal understanding of the world. Knowledge graphs that represent structural relations between entities have become an increasingly popular research direction towards cognition and human-level intelligence. In this survey, we provide a comprehensive review of knowledge graph covering overall research topics about 1) knowledge graph representation learning, 2) knowledge acquisition and completion, 3) temporal knowledge graph, and 4) knowledge-aware applications, and summarize recent breakthroughs and perspective directions to facilitate future research. We propose a full-view categorization and new taxonomies on these topics. Knowledge graph embedding is organized from four aspects of representation space, scoring function, encoding models, and auxiliary information. For knowledge acquisition, especially knowledge graph completion, embedding methods, path inference, and logical rule reasoning, are reviewed. We further explore several emerging topics, including meta relational learning, commonsense reasoning, and temporal knowledge graphs. To facilitate future research on knowledge graphs, we also provide a curated collection of datasets and open-source libraries on different tasks. In the end, we have a thorough outlook on several promising research directions.**

***Index Terms*—Knowledge graph, representation learning, knowledge graph completion, relation extraction, reasoning, deep learning.**

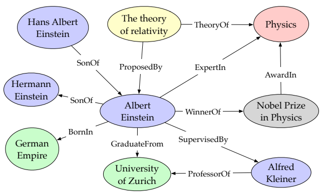
I. Introduction

INCORPORATING human knowledge is one of the research directions of artificial intelligence (AI). Knowledge representation and reasoning, inspired by human problem solving, is to represent knowledge for intelligent systems to gain the ability to solve complex tasks [1], [2]. Recently, knowledge graphs as a form of structured human knowledge have drawn great research attention from both the academia and the industry [3]–[6]. A knowledge graph is a structured representation of facts, consisting of entities, relationships, and semantic descriptions. Entities can be real-world objects and abstract concepts, relationships represent the relation between entities, and semantic descriptions of entities, and their relationships contain types and properties with a well-defined meaning. Property graphs or attributed graphs are widely used, in which nodes and relations have properties or attributes.

The term of knowledge graph is synonymous with knowledge base with a minor difference. A knowledge graph can be viewed as a graph when considering its graph structure [7]. When it involves formal semantics, it can be taken as a knowledge base for interpretation and inference over facts [8]. Examples of knowledge base and knowledge graph are illustrated in Fig. 1. Knowledge can be expressed in a factual triple in the form of (*head*, `relation`, *tail*) or (*subject*, `predicate`, *object*) under the resource description framework (RDF), for example, (*Albert Einstein*, `WinnerOf`, *Nobel Prize*). It can also be represented as a directed graph with nodes as entities and edges as relations. For simplicity and following the trend of the research community, this paper uses the terms knowledge graph and knowledge base interchangeably.

(Albert Einstein, **BornIn**, German Empire)
(Albert Einstein, **SonOf**, Hermann Einstein)
(Albert Einstein, **GraduateFrom**, University of Zurich)
(Albert Einstein, **WinnerOf**, Nobel Prize in Physics)
(Albert Einstein, **ExpertIn**, Physics)
(Nobel Prize in Physics, **AwardIn**, Physics)
(The theory of relativity, **TheoryOf**, Physics)
(Albert Einstein, **SupervisedBy**, Alfred Kleiner)
(Alfred Kleiner, **ProfessorOf**, University of Zurich)
(The theory of relativity, **ProposedBy**, Albert Einstein)
(Hans Albert Einstein, **SonOf**, Albert Einstein)

(a) Factual triples in knowledge base.



(b) Entities and relations in knowledge graph.

Fig. 1: An example of knowledge base and knowledge graph.

Recent advances in knowledge-graph-based research focus on knowledge representation learning (KRL) or knowledge graph embedding (KGE) by mapping entities and relations into low-dimensional vectors while capturing their semantic meanings [5], [9]. Specific knowledge acquisition tasks include knowledge graph completion (KGC), triple classification, entity recognition, and relation extraction. Knowledge-aware models benefit from the integration of heterogeneous information, rich ontologies and semantics for knowledge representation, and multi-lingual knowledge. Thus, many real-world applications such as recommendation systems and question answering have been brought about prosperity with the ability of commonsense understanding and reasoning. Some real-world products, for example, Microsoft's Satori and Google's Knowledge Graph [3], have shown a strong capacity to provide more efficient services.

This paper conducts a comprehensive survey of current literature on knowledge graphs, which enriches graphs with more context, intelligence, and semantics for knowledge acquisition and knowledge-aware applications. Our main contributions are summarized as follows.

- **Comprehensive review.** We conduct a comprehensive review of the origin of knowledge graph and modern techniques for relational learning on knowledge graphs.

S. Ji and P. Marttinen are with Aalto University, Finland. E-mail: {shaoxiong.ji; pekka.marttinen}@aalto.fi

S. Pan is with Monash University, Australia. E-mail: shirui.pan@monash.edu

E. Cambria is with Nanyang Technological University, Singapore. E-mail: cambria@ntu.edu.sg

P.S. Yu is with University of Illinois at Chicago, USA. E-mail: psyu@uic.edu

S. Pan is the corresponding author.

Major neural architectures of knowledge graph representation learning and reasoning are introduced and compared. Moreover, we provide a complete overview of many applications in different domains.

- **Full-view categorization and new taxonomies.** A full-view categorization of research on knowledge graph, together with fine-grained new taxonomies are presented. Specifically, in the high-level, we review the research on knowledge graphs in four aspects: KRL, knowledge acquisition, temporal knowledge graphs, and knowledge-aware application. For KRL approaches, we further propose fine-grained taxonomies into four views, including representation space, scoring function, encoding models, and auxiliary information. For knowledge acquisition, KGC is reviewed under embedding-based ranking, relational path reasoning, logical rule reasoning, and meta relational learning; entity-relation acquisition tasks are divided into entity recognition, typing, disambiguation, and alignment; and relation extraction is discussed according to the neural paradigms.
- **Wide coverage on emerging advances.** Knowledge graph has experienced rapid development. This survey provides wide coverage on emerging topics, including transformer-based knowledge encoding, graph neural network (GNN) based knowledge propagation, reinforcement learning-based path reasoning, and meta relational learning.
- **Summary and outlook on future directions.** This survey provides a summary of each category and highlights promising future research directions.

The remainder of this survey is organized as follows: first, an overview of knowledge graphs including history, notations, definitions, and categorization is given in Section II; then, we discuss KRL in Section III from four scopes; next, our review goes to tasks of knowledge acquisition and temporal knowledge graphs in Section IV and Section V; downstream applications are introduced in Section VI; finally, we discuss future research directions, together with a conclusion in the end. Other information, including KRL model training and a collection of knowledge graph datasets and open-source implementations, can be found in the appendices.

II. Overview

A. A Brief History of Knowledge Bases

Knowledge representation has experienced a long-period history of development in the fields of logic and AI. The idea of graphical knowledge representation firstly dated back to 1956 as the concept of semantic net proposed by Richens [10], while the symbolic logic knowledge can go back to the General Problem Solver [1] in 1959. The knowledge base is firstly used with knowledge-based systems for reasoning and problem-solving. MYCIN [2] is one of the most famous rule-based expert systems for medical diagnosis with a knowledge base of about 600 rules. Later, the community of human knowledge representation saw the development of frame-based language, rule-based, and hybrid representations. Approximately at the end of this period, the Cyc project[1] began, aiming at assembling human knowledge. Resource description framework (RDF)[2] and Web Ontology Language (OWL)[3] were released in turn, and became important standards of the Semantic Web[4]. Then, many open knowledge bases or ontologies were published, such as WordNet, DBpedia, YAGO, and Freebase. Stokman and Vries [7] proposed a modern idea of structure knowledge in a graph in 1988. However, it was in 2012 that the concept of knowledge graph gained great popularity since its first launch by Google's search engine[5], where the knowledge fusion framework called Knowledge Vault [3] was proposed to build large-scale knowledge graphs. A brief road map of knowledge base history is illustrated in Fig. 2.

B. Definitions and Notations

Most efforts have been made to give a definition by describing general semantic representation or essential characteristics. However, there is no such wide-accepted formal definition. Paulheim [11] defined four criteria for knowledge graphs. Ehrlinger and Wöß [12] analyzed several existing definitions and proposed Definition 1, which emphasizes the reasoning engine of knowledge graphs. Wang et al. [5] proposed a definition as a multi-relational graph in Definition 2. Following previous literature, we define a knowledge graph as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{F}$ are sets of entities, relations and facts, respectively. A fact is denoted as a triple $(h, r, t) \in \mathcal{F}$.

Definition 1 (Ehrlinger and Wöß [12])**.** A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge.

Definition 2 (Wang et al. [5])**.** A knowledge graph is a multi-relational graph composed of entities and relations which are regarded as nodes and different types of edges, respectively.

Specific notations and their descriptions are listed in Table I. Details of several mathematical operations are explained in Appendix A.

C. Categorization of Research on Knowledge Graph

This survey provides a comprehensive literature review on the research of knowledge graphs, namely KRL, knowledge acquisition, and a wide range of downstream knowledge-aware applications, where many recent advanced deep learning techniques are integrated. The overall categorization of the research is illustrated in Fig. 3.

Knowledge Representation Learning is a critical research issue of knowledge graph which paves the way for many knowledge acquisition tasks and downstream applications. We categorize KRL into four aspects of *representation space*, *scoring function*, *encoding models* and *auxiliary information*,

[1] http://cyc.com

[2] Released as W3C recommendation in 1999 available at http://w3.org/TR/1999/REC-rdf-syntax-19990222.

[3] http://w3.org/TR/owl-guide

[4] http://w3.org/standards/semanticweb

[5] http://blog.google/products/search/introducing-knowledge-graph-things-not

Fig. 2: A brief history of knowledge bases

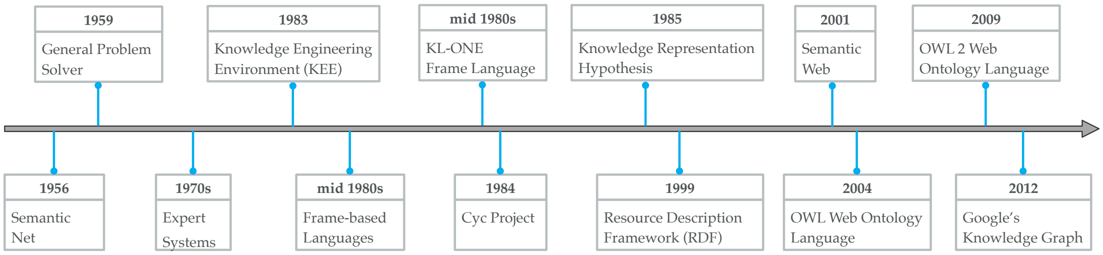


TABLE I: Notations and descriptions.

Notation	Description
$\mathcal{G}$	A knowledge graph
$\mathcal{F}$	A set of facts
(h, r, t)	A triple of head, relation and tail
$(\mathbf{h}, \mathbf{r}, \mathbf{t})$	Embedding of head, relation and tail
$r \in \mathcal{R}, e \in \mathcal{E}$	Relation set and entity set
$v \in \mathcal{V}$	Vertex in vertice set
$\xi \in \mathcal{E}_{\mathcal{G}}$	Edge in edge set
e_s, e_q, e_t	Source/query/current entity
r_q	Query relation
$< w_1, \ldots, w_n >$	Text corpus
$d.(\cdot)$	Distance metric in specific space
$f_r(\mathbf{h}, \mathbf{t})$	Scoring function
$\sigma(\cdot)$, $g(\cdot)$	Non-linear activation function
$\mathbf{M}_r$	Mapping matrix
$\widehat{\mathbf{M}}$	Tensor
$\mathcal{L}$	Loss function
$\mathbb{R}^d$	d dimensional real-valued space
$\mathbb{C}^d$	d dimensional complex space
$\mathbb{H}^d$	d dimensional hypercomplex space
$\mathbb{T}^d$	d dimensional torus space
$\mathbb{B}^d_c$	d dimensional hyperbolic space with curvature c
$\mathcal{N}(\mathbf{u}, \sigma^2\mathbf{I})$	Gaussian distribution
$\langle \mathbf{h}, \mathbf{t} \rangle$	Hermitian dot product
$\mathbf{t} \otimes \mathbf{r}$	Hamilton product
$\mathbf{h} \circ \mathbf{t}$, $\mathbf{h} \odot \mathbf{t}$	Hadmard (element-wise) product
$\mathbf{h} \star \mathbf{t}$	Circular correlation
concat(), $[\mathbf{h}, \mathbf{r}]$	Vectors/matrices concatenation
$\boldsymbol{\omega}$	Convolutional filters
$*$	Convolution operator

providing a clear workflow for developing a KRL model. Specific ingredients include:

1) *representation space* in which the relations and entities are represented;
2) *scoring function* for measuring the plausibility of factual triples;
3) *encoding models* for representing and learning relational interactions;
4) *auxiliary information* to be incorporated into the embedding methods.

Representation learning includes point-wise space, manifold, complex vector space, Gaussian distribution, and discrete space. Scoring metrics are generally divided into distance-based and similarity matching based scoring functions. Current research focuses on encoding models, including linear/bilinear models, factorization, and neural networks. Auxiliary information considers textual, visual, and type information.

Knowledge Acquisition tasks are divided into three categories, i.e., KGC, relation extraction, and entity discovery. The first one is for expanding existing knowledge graphs, while the other two discover new knowledge (aka relations and entities) from the text. KGC falls into the following categories: embedding-based ranking, relation path reasoning, rule-based reasoning, and meta relational learning. Entity discovery includes recognition, disambiguation, typing, and alignment. Relation extraction models utilize attention mechanism, graph convolutional networks (GCNs), adversarial training, reinforcement learning, deep residual learning, and transfer learning.

Temporal Knowledge Graphs incorporate temporal information for representation learning. This survey categorizes four research fields, including temporal embedding, entity dynamics, temporal relational dependency, and temporal logical reasoning.

Knowledge-aware Applications include natural language understanding (NLU), question answering, recommendation systems, and miscellaneous real-world tasks, which inject knowledge to improve representation learning.

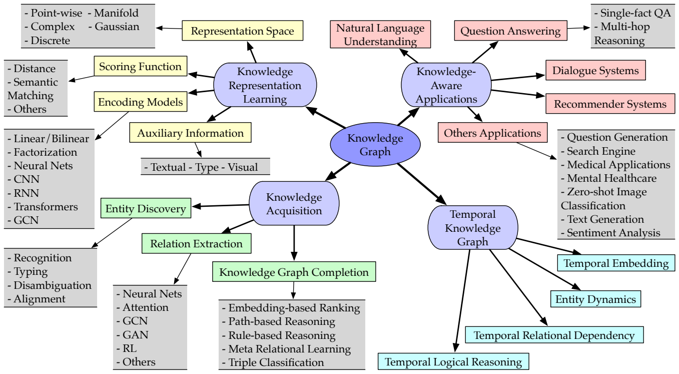


Fig. 3: Categorization of research on knowledge graphs.

D. Related Surveys

Previous survey papers on knowledge graphs mainly focus on statistical relational learning [4], knowledge graph refinement [11], Chinese knowledge graph construction [13], knowledge reasoning [14], KGE [5] or KRL [9]. The latter two surveys are more related to our work. Lin et al. [9] presented KRL in a linear manner, with a concentration on quantitative analysis. Wang et al. [5] categorized KRL according to scoring functions and specifically focused on the type of information utilized in KRL. It provides a general view of current research only from the perspective of scoring metrics. Our survey goes deeper to the flow of KRL and provides a full-scaled view from four-folds, including representation space, scoring function, encoding models, and auxiliary information. Besides, our paper provides a comprehensive review of knowledge acquisition and knowledge-aware applications with several emerging topics such as knowledge-graph-based reasoning and few-shot learning discussed.

III. Knowledge Representation Learning

KRL is also known as KGE, multi-relation learning, and statistical relational learning in the literature. This section reviews recent advances on distributed representation learning with rich semantic information of entities and relations form four scopes including representation space (representing entities and relations, **Section III-A**), scoring function (measuring the plausibility of facts, **Section III-B**), encoding models (modeling the semantic interaction of facts, **Section III-C**), and auxiliary information (utilizing external information, **Section III-D**). We further provide a summary in **Section III-E**. The training strategies for KRL models are reviewed in Appendix C.

A. Representation Space

The key issue of representation learning is to learn low-dimensional distributed embedding of entities and relations. Current literature mainly uses real-valued point-wise space (Fig. 4a) including vector, matrix and tensor space, while other kinds of space such as complex vector space (Fig. 4b), Gaussian space (Fig. 4c), and manifold (Fig. 4d) are utilized as well.

1) Point-Wise Space: Point-wise Euclidean space is widely applied for representing entities and relations, projecting relation embedding in vector or matrix space, or capturing relational interactions. TransE [15] represents entities and relations in d-dimension vector space, i.e., $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$, and makes embeddings follow the translational principle $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$. To tackle this problem of insufficiency of a single space for both entities and relations, TransR [16] then further introduces separated spaces for entities and relations. The authors projected entities ($\mathbf{h}, \mathbf{t} \in \mathbb{R}^k$) into relation ($\mathbf{r} \in \mathbb{R}^d$) space by a projection matrix $\mathbf{M_r} \in \mathbb{R}^{k \times d}$. NTN [17] models entities across multiple dimensions by a bilinear tensor neural layer. The relational interaction between head and tail $\mathbf{h}^T \widehat{\mathbf{M}} \mathbf{t}$ is captured as a tensor denoted as $\widehat{\mathbf{M}} \in \mathbb{R}^{d \times d \times k}$. Instead of using the Cartesian coordinate system, HAKE [18] captures semantic hierarchies by mapping entities into the polar coordinate system, i.e., entity embeddings $\mathbf{e}_m \in \mathbb{R}^d$ and $\mathbf{e}_p \in [0, 2\pi)^d$ in the modulus and phase part, respectively.

Many other translational models such as TransH [19] also use similar representation space, while semantic matching models use plain vector space (e.g., HolE [20]) and relational projection matrix (e.g., ANALOGY [21]). Principles of these translational and semantic matching models are introduced in Section III-B1 and III-B2, respectively.

2) Complex Vector Space: Instead of using a real-valued space, entities and relations are represented in a complex space, where $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{C}^d$. Take head entity as an example, $\mathbf{h}$ has a real part $\mathrm{Re}(\mathbf{h})$ and an imaginary part $\mathrm{Im}(\mathbf{h})$, i.e., $\mathbf{h} = \mathrm{Re}(\mathbf{h}) + i\,\mathrm{Im}(\mathbf{h})$. ComplEx [22] firstly introduces complex vector space shown in Fig. 4b which can capture both symmetric and antisymmetric relations. Hermitian dot product is used to do composition for relation, head and the conjugate of tail. Inspired by Euler's identity $e^{i\theta} = \cos\theta + i \sin\theta$, RotatE [23] proposes a rotational model taking relation as a rotation from head entity to tail entity in complex space as $\mathbf{t} = \mathbf{h} \circ \mathbf{r}$ where $\circ$ denotes the element-wise Hadmard product. QuatE [24] extends the complex-valued space into hypercomplex $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{H}^d$ by a quaternion $Q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ with three imaginary components, where the quaternion inner product, i.e., the Hamilton product $\mathbf{h} \otimes \mathbf{r}$, is used as compositional operator for head entity and relation.

3) Gaussian Distribution: Inspired by Gaussian word embedding, the density-based embedding model KG2E [25] introduces Gaussian distribution to deal with the (un)certainties of entities and relations. The authors embedded entities and relations into multi-dimensional Gaussian distribution $\mathcal{H} \sim \mathcal{N}(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h)$ and $\mathcal{T} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. The mean vector $\mathbf{u}$ indicates entities and relations' position, and the covariance matrix $\boldsymbol{\Sigma}$ models their (un)certainties. Following the translational principle, the probability distribution of entity transformation $\mathcal{H} - \mathcal{T}$ is denoted as $\mathcal{P}_e \sim \mathcal{N}(\boldsymbol{\mu}_h - \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_h + \boldsymbol{\Sigma}_t)$. Similarly, TransG [26] represents entities with Gaussian distributions, while it draws a mixture of Gaussian distribution for relation embedding, where the m-th component translation vector of relation r is denoted as $\mathbf{u}_{r,m} = \mathbf{t} - \mathbf{h} \sim \mathcal{N}\left(\mathbf{u_t} - \mathbf{u_h}, \left(\sigma_h^2 + \sigma_t^2\right)\mathbf{E}\right)$.

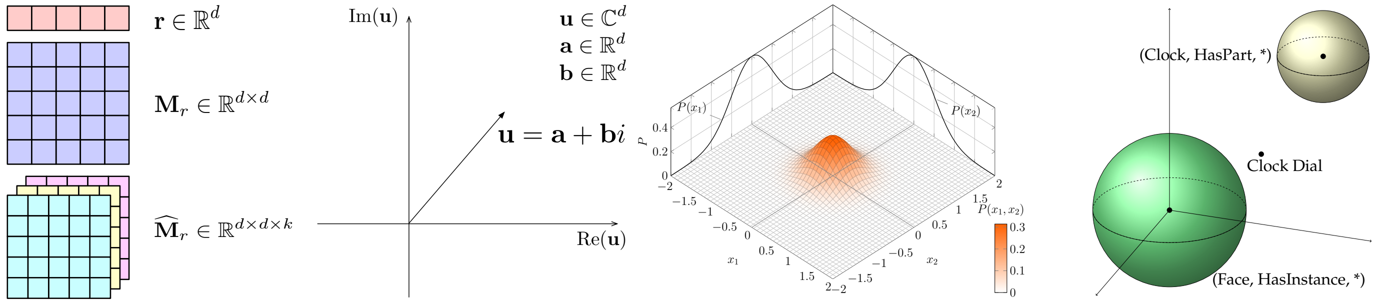
4) Manifold and Group: This section reviews knowledge representation in manifold space, Lie group, and dihedral group. A manifold is a topological space, which could be defined as a set of points with neighborhoods by the set theory. The group is algebraic structures defined in abstract algebra. Previous point-wise modeling is an ill-posed algebraic system where the number of scoring equations is far more than the number of entities and relations. Moreover, embeddings are restricted in an overstrict geometric form even in some methods with subspace projection. To tackle these issues, ManifoldE [27] extends point-wise embedding into manifold-based embedding. The authors introduced two settings of manifold-based embedding, i.e., Sphere and Hyperplane. An example of a sphere is shown in Fig. 4d. For the sphere setting, Reproducing Kernel Hilbert Space is used to represent the manifold function, i.e.,

$$\begin{aligned} \mathcal{M}(h, r, t) &= \|\varphi(h) + \varphi(r) - \varphi(t)\|^2 \\ &= \mathbf{K}(h, h) + \mathbf{K}(t, t) + \mathbf{K}(r, r) \\ &\quad - 2\mathbf{K}(h, t) - 2\mathbf{K}(r, t) + 2\mathbf{K}(r, h), \end{aligned} \tag{1}$$

where φ maps the original space to the Hilbert space, and $\mathbf{K}$ is the kernel function. Another "hyperplane" setting is introduced to enhance the model with intersected embeddings, i.e.,

$$\mathcal{M}(h, r, t) = \left(\mathbf{h} + \mathbf{r}_{\text{ head}}\right)^{\top} \left(\mathbf{t} + \mathbf{r}_{\text{ tail}}\right). \tag{2}$$

Hyperbolic space, a multidimensional Riemannian manifold with a constant negative curvature $-c$ $(c > 0)$: $\mathbb{B}^{d,c} = \left\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|^2 < \frac{1}{c}\right\}$, is drawing attention for its capacity of capturing hierarchical information. MuRP [28] represents the multi-relational knowledge graph in Poincar ball of hyperbolic space $\mathbb{B}_c^d = \left\{\mathbf{x} \in \mathbb{R}^d : c\|\mathbf{x}\|^2 < 1\right\}$. While it fails to capture logical patterns and suffers from constant curvature. Chami et


$\mathbf{r} \in \mathbb{R}^d$
$\mathbf{M}_r \in \mathbb{R}^{d \times d}$
$\widehat{\mathbf{M}}_r \in \mathbb{R}^{d \times d \times k}$
Im(u)
Re(u)
$\mathbf{u} \in \mathbb{C}^d$
$\mathbf{a} \in \mathbb{R}^d$
$\mathbf{b} \in \mathbb{R}^d$
$\mathbf{u} = \mathbf{a} + \mathbf{b}i$
$P(x_1)$
$P(x_2)$
$P(x_1, x_2)$
x_1
x_2
(Clock, HasPart, *)
Clock Dial
(Face, HasInstance, *)

(a) Point-wise space. (b) Complex vector space. (c) Gaussian distribution. (d) Manifold space.

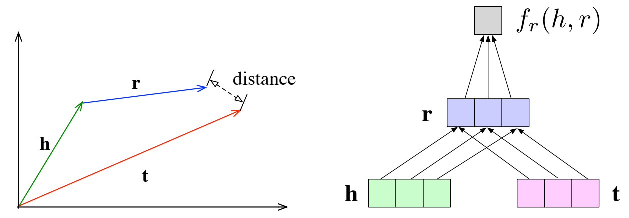
Fig. 4: An illustration of knowledge representation in different spaces.

al. [29] leverages expressive hyperbolic isometries and learns a relation-specific absolute curvature c_r in the hyperbolic space.

TorusE [30] solves the regularization problem of TransE via embedding in an n-dimensional torus space which is a compact Lie group. With the projection from vector space into torus space defined as $\pi : \mathbb{R}^n \to T^n, x \mapsto [x]$, entities and relations are denoted as $[\mathbf{h}], [\mathbf{r}], [\mathbf{t}] \in \mathbb{T}^n$. Similar to TransE, it also learns embeddings following the relational translation in torus space, i.e., $[\mathbf{h}] + [\mathbf{r}] \approx [\mathbf{t}]$. Recently, DihEdral [31] proposes a dihedral symmetry group preserving a 2-dimensional polygon.

B. Scoring Function

The scoring function is used to measure the plausibility of facts, also referred to as the energy function in the energy-based learning framework. Energy-based learning aims to learn the energy function $\mathcal{E}_\theta(x)$ (parameterized by θ taking x as input) and to make sure positive samples have higher scores than negative samples. In this paper, the term of the scoring function is adopted for unification. There are two typical types of scoring functions, i.e., distance-based (Fig. 5a) and similarity-based (Fig. 5b) functions, to measure the plausibility of a fact. Distance-based scoring function measures the plausibility of facts by calculating the distance between entities, where addictive translation with relations as $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$ is widely used. Semantic similarity based scoring measures the plausibility of facts by semantic matching. It usually adopts multiplicative formulation, i.e., $\mathbf{h}^\top \mathbf{M}_r \approx \mathbf{t}^\top$, to transform head entity near the tail in the representation space.



(a) Translational distance-based scoring of TransE. (b) Semantic similarity-based scoring of DistMult.

Fig. 5: Illustrations of distance-based and similarity matching based scoring functions taking TransE [15] and DistMult [32] as examples.

1) Distance-based Scoring Function: An intuitive distance-based approach is to calculate the Euclidean distance between the relational projection of entities. Structural Embedding (SE) [8] uses two projection matrices and L_1 distance to learn structural embedding as

$$f_r(h,t) = \|\mathbf{M}_{r,1}\mathbf{h} - \mathbf{M}_{r,2}\mathbf{t}\|_{L_1}. \tag{3}$$

A more intensively used principle is the translation-based scoring function that aims to learn embeddings by representing relations as translations from head to tail entities. Bordes et al. [15] proposed TransE by assuming that the added embedding of $\mathbf{h}+\mathbf{r}$ should be close to the embedding of $\mathbf{t}$ with the scoring function is defined under L_1 or L_2 constraints as

$$f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_1/L_2}. \tag{4}$$

Since that, many variants and extensions of TransE have been proposed. For example, TransH [19] projects entities and relations into a hyperplane, TransR [16] introduces separate projection spaces for entities and relations, and TransD [33] constructs dynamic mapping matrices $\mathbf{M}_{rh} = \mathbf{r}_p\mathbf{h}_p^\top + \mathbf{I}$ and $\mathbf{M}_{rt} = \mathbf{r}_p\mathbf{t}_p^\top + \mathbf{I}$ by the projection vectors $\mathbf{h}_p, \mathbf{t}_p, \mathbf{r}_p \in \mathbb{R}^n$. By replacing Euclidean distance, TransA [34] uses Mahalanobis distance to enable more adaptive metric learning. Previous methods used additive score functions, TransF [35] relaxes the strict translation and uses dot product as $f_r(h,t) = (\mathbf{h}+\mathbf{r})^\top\mathbf{t}$. To balance the constraints on head and tail, a flexible translation scoring function is further proposed.

Recently, ITransF [36] enables hidden concepts discovery and statistical strength transferring by learning associations between relations and concepts via sparse attention vectors, with scoring function defined as

$$f_r(h,t) = \left\| \boldsymbol{\alpha}_r^H \cdot \mathbf{D} \cdot \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_r^T \cdot \mathbf{D} \cdot \mathbf{t} \right\|_\ell, \tag{5}$$

where $\mathbf{D} \in \mathbb{R}^{n \times d \times d}$ is stacked concept projection matrices of entities and relations and $\boldsymbol{\alpha}_r^H, \boldsymbol{\alpha}_r^T \in [0,1]^n$ are attention vectors calculated by sparse softmax, TransAt [37] integrates relation attention mechanism with translational embedding, and TransMS [38] transmits multi-directional semantics with nonlinear functions and linear bias vectors, with the scoring function as

$$f_r(\mathbf{h},\mathbf{t}) = \|{-\tanh(\mathbf{t}\circ\mathbf{r})\circ\mathbf{h}} + \mathbf{r} - \tanh(\mathbf{h}\circ\mathbf{r})\circ\mathbf{t} + \alpha\cdot(\mathbf{h}\circ\mathbf{t})\|_{\ell_{1/2}}. \tag{6}$$

KG2E [25] in Gaussian space and ManifoldE [27] with manifold also use the translational distance-based scoring function. KG2E uses two scoring methods, i.e, asymmetric KL-divergence and symmetric expected likelihood. While the scoring function of ManifoldE is defined as

$$f_r(h,t) = \left\|\mathcal{M}(h,r,t) - D_r^2\right\|^2, \tag{7}$$

where $\mathcal{M}$ is the manifold function, and D_r is a relation-specific manifold parameter.

2) Semantic Matching: Another direction is to calculate the semantic similarity. SME [39] proposes to semantically match separate combinations of entity-relation pairs of (h,r) and (r,t). Its scoring function is defined with two versions of matching blocks - linear and bilinear block, i.e.,

$$f_r(h,t) = g_{\text{left}}(\mathbf{h},\mathbf{r})^\top g_{\text{right}}(\mathbf{r},\mathbf{t}). \tag{8}$$

The linear matching block is defined as $g_{\text{left}}(h,t) = \mathbf{M}_{l,1}\mathbf{h}^\top + \mathbf{M}_{l,2}\mathbf{r}^\top + \mathbf{b}_l^\top$, and the bilinear form is $g_{\text{left}}(\mathbf{h},\mathbf{r}) = (\mathbf{M}_{l,1}\mathbf{h}) \circ (\mathbf{M}_{l,2}\mathbf{r}) + \mathbf{b}_l^\top$. By restricting relation matrix M_r to be diagonal for multi-relational representation learning, DistMult [32] proposes a simplified bilinear formulation defined as

$$f_r(h,t) = \mathbf{h}^\top \operatorname{diag}(\mathbf{M}_r)\mathbf{t}. \tag{9}$$

To capture productive interactions in relational data and compute efficiently, HolE [20] introduces a circular correlation of embedding, which can be interpreted as a compressed tensor product, to learn compositional representations. By defining a perturbed holographic compositional operator as $p(\boldsymbol{a},\boldsymbol{b};\boldsymbol{c}) = (\boldsymbol{c}\circ\boldsymbol{a})\star\boldsymbol{b}$, where $\mathbf{c}$ is a fixed vector, the expanded holographic embedding model HolEx [40] interpolates the HolE and full tensor product method. It can be viewed as linear concatenation of perturbed HolE. Focusing on multi-relational inference, ANALOGY [21] models analogical structures of relational data. It's scoring function is defined as

$$f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t}, \tag{10}$$

with relation matrix constrained to be normal matrices in linear mapping, i.e., $\mathbf{M}_r^\top \mathbf{M}_r = \mathbf{M}_r \mathbf{M}_r^\top$ for analogical inference. Crossover interactions are introduced by CrossE [41] with an interaction matrix $\mathbf{C} \in \mathbb{R}^{n_r \times d}$ to simulate the bi-directional interaction between entity and relation. The relation specific interaction is obtained by looking up interaction matrix as $\mathbf{c}_r = \mathbf{x}_r^\top \mathbf{C}$. By combining the interactive representations and matching with tail embedding, the scoring function is defined as

$$f(h,r,t) = \sigma\left(\tanh\left(\mathbf{c}_r \circ \mathbf{h} + \mathbf{c}_r \circ \mathbf{h} \circ \mathbf{r} + \mathbf{b}\right)\mathbf{t}^\top\right). \tag{11}$$

The semantic matching principle can be encoded by neural networks further discussed in Sec. III-C.

The two methods mentioned above in Sec. III-A4 with group representation also follow the semantic matching principle. The scoring function of TorusE [30] is defined as:

$$\min_{(x,y)\in([h]+[r])\times[t]} \|x-y\|_i. \tag{12}$$

By modeling $2L$ relations as group elements, the scoring function of DihEdral [31] is defined as the summation of components:

$$f_r(h,t) = \mathbf{h}^\top \mathbf{R} \mathbf{t} = \sum_{l=1}^{L} \mathbf{h}^{(l)\top} \mathbf{R}^{(l)} \mathbf{t}^{(l)}, \tag{13}$$

where the relation matrix $\mathbf{R}$ is defined in block diagonal form for $\mathbf{R}^{(l)} \in \mathbb{D}_K$, and entities are embedded in real-valued space for $\mathbf{h}^{(l)}$ and $\mathbf{t}^{(l)} \in \mathbb{R}^2$.

C. Encoding Models

This section introduces models that encode the interactions of entities and relations through specific model architectures, including linear/bilinear models, factorization models, and neural networks. Linear models formulate relations as a linear/bilinear mapping by projecting head entities into a representation space close to tail entities. Factorization aims to decompose relational data into low-rank matrices for representation learning. Neural networks encode relational data with non-linear neural activation and more complex network structures. Several neural models are illustrated in Fig. 6.

1) Linear/Bilinear Models: Linear/bilinear models encode interactions of entities and relations by applying linear operation as:

$$g_r(\mathbf{h},\mathbf{t}) = \mathbf{M}_r^T \begin{pmatrix} \mathbf{h} \\ \mathbf{t} \end{pmatrix}, \tag{14}$$

or bilinear transformation operations as Eq. 10. Canonical methods with linear/bilinear encoding include SE [8], SME [39], DistMult [32], ComplEx [22], and ANALOGY [21]. For TransE [15] with L2 regularization, the scoring function can be expanded to the form with only linear transformation with one-dimensional vectors, i.e.,

$$\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2 = 2\mathbf{r}^T(\mathbf{h}-\mathbf{t}) - 2\mathbf{h}^T\mathbf{t} + \|\mathbf{r}\|_2^2 + \|\mathbf{h}\|_2^2 + \|\mathbf{t}\|_2^2. \tag{15}$$

Wang et al. [46] studied various bilinear models and evaluated their expressiveness and connections by introducing the concepts of universality and consistency. The authors further showed that the ensembles of multiple linear models can improve the prediction performance through experiments. Recently, to solve the independence embedding issue of entity vectors in canonical Polyadia decomposition, SimplE [47] introduces the inverse of relations and calculates the average canonical Polyadia score of (h,r,t) and (t,r^{-1},h) as

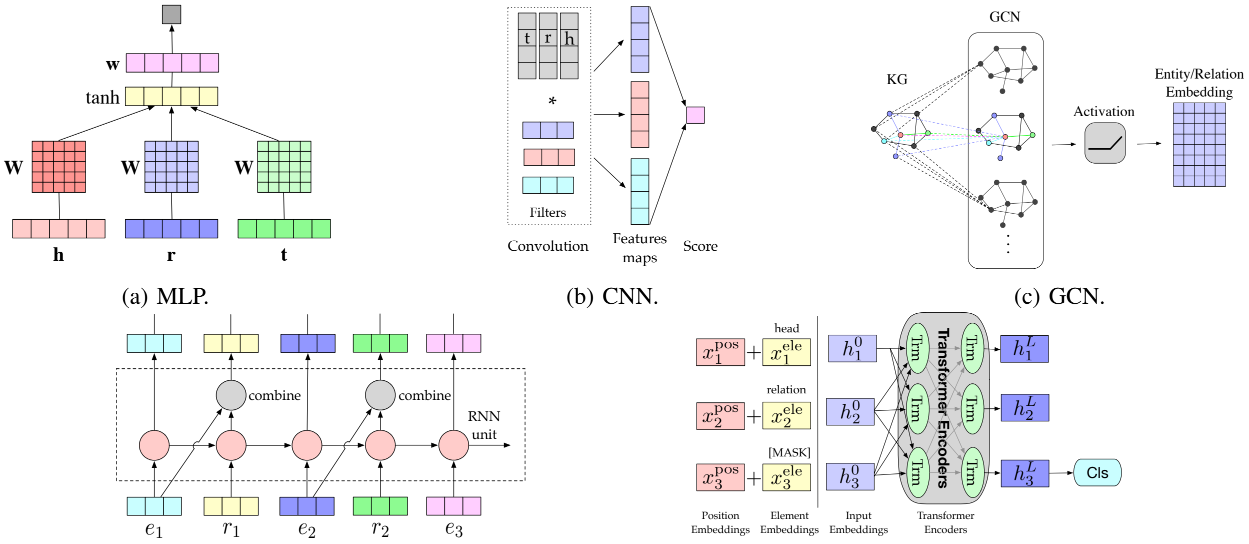
$$f_r(h,t) = \frac{1}{2}\left(\mathbf{h}\circ\mathbf{r}\mathbf{t} + \mathbf{t}\circ\mathbf{r}'\mathbf{t}\right), \tag{16}$$

where $\mathbf{r}'$ is the embedding of inversion relation. More bilinear models are proposed from a factorization perspective discussed in the next section.

2) Factorization Models: Factorization methods formulated KRL models as three-way tensor $\mathcal{X}$ decomposition. A general principle of tensor factorization can be denoted as $\mathcal{X}_{hrt} \approx \mathbf{h}^\top \mathbf{M}_r \mathbf{t}$, with the composition function following the semantic matching pattern. Nickel et al. [48] proposed the three-way rank-r factorization RESCAL over each relational slice of knowledge graph tensor. For k-th relation of m relations, the k-th slice of $\mathcal{X}$ is factorized as

$$\mathcal{X}_k \approx \mathbf{A}\mathbf{R}_k\mathbf{A}^T. \tag{17}$$

The authors further extended it to handle attributes of entities efficiently [49]. Jenatton et al. [50] then proposed a bilinear structured latent factor model (LFM), which extends RESCAL by decomposing $\mathbf{R}_k = \sum_{i=1}^{d} \boldsymbol{\alpha}_i^k \mathbf{u}_i \mathbf{v}_i^\top$. By introducing three-way Tucker tensor decomposition, TuckER [51] learns to



(a) MLP.

(b) CNN.

(c) GCN.

(d) RSN.

(e) Transformer.

Fig. 6: Illustrations of neural encoding models. (a) MLP [3] and (b) CNN [42] input triples into dense layer and convolution operation to learn semantic representation, (c) GCN [43] acts as encoder of knowledge graphs to produce entity and relation embeddings. (d) RSN [44] encodes entity-relation sequences and skips relations discriminatively. (e) Transformer-based CoKE [45] encodes triples as sequences with an entity replaced by `[MASK]`.

embed by outputting a core tensor and embedding vectors of entities and relations. LowFER [52] proposes a multimodal factorized bilinear pooling mechanism to better fuse entities and relations. It generalizes the TuckER model and is computationally efficient with low-rank approximation.

3) Neural Networks: Neural networks for encoding semantic matching have yielded remarkable predictive performance in recent studies. Encoding models with linear/bilinear blocks can also be modeled using neural networks, for example, SME [39]. Representative neural models include multi-layer perceptron (MLP) [3], neural tensor network (NTN) [17], and neural association model (NAM) [53]. They generally take entities or relations or both of them into deep neural networks and compute a semantic matching score. MLP [3] (Fig. 6a) encodes entities and relations together into a fully-connected layer, and uses a second layer with sigmoid activation for scoring a triple as

$$f_r(h,t) = \sigma(\mathbf{w}^\top \sigma(\mathbf{W}[\mathbf{h},\mathbf{r},\mathbf{t}])), \tag{18}$$

where $\mathbf{W} \in \mathbb{R}^{n\times 3d}$ is the weight matrix and $[\mathbf{h},\mathbf{r},\mathbf{t}]$ is a concatenation of three vectors. NTN [17] takes entity embeddings as input associated with a relational tensor and outputs predictive score in as

$$f_r(h,t) = \mathbf{r}^\top \sigma(\mathbf{h}^T \widehat{\mathbf{M}}\mathbf{t} + \mathbf{M}_{r,1}\mathbf{h} + \mathbf{M}_{r,2}\mathbf{t} + \mathbf{b}_r), \tag{19}$$

where $\mathbf{b}_r \in \mathbb{R}^k$ is bias for relation r, $\mathbf{M}_{r,1}$ and $\mathbf{M}_{r,2}$ are relation-specific weight matrices. It can be regarded as a combination of MLPs and bilinear models. NAM [53] associates the hidden encoding with the embedding of tail entity, and proposes the relational-modulated neural network (RMNN).

4) Convolutional Neural Networks: CNNs are utilized for learning deep expressive features. ConvE [54] uses 2D convolution over embeddings and multiple layers of nonlinear features to model the interactions between entities and relations by reshaping head entity and relation into 2D matrix, i.e., $\mathbf{M}_h \in \mathbb{R}^{d_w \times d_h}$ and $\mathbf{M}_r \in \mathbb{R}^{d_w \times d_h}$ for $d = d_w \times d_h$. Its scoring function is defined as

$$f_r(h,t) = \sigma\left(\mathrm{vec}\left(\sigma\left([\mathbf{M}_h;\mathbf{M}_r] * \boldsymbol{\omega}\right)\right)\mathbf{W}\right)\mathbf{t}, \tag{20}$$

where $\boldsymbol{\omega}$ is the convolutional filters and vec is the vectorization operation reshaping a tensor into a vector. ConvE can express semantic information by non-linear feature learning through multiple layers. ConvKB [42] adopts CNNs for encoding the concatenation of entities and relations without reshaping (Fig. 6b). Its scoring function is defined as

$$f_r(h,t) = \mathrm{concat}\left(\sigma\left([\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}] * \boldsymbol{\omega}\right)\right)\cdot \mathbf{w}. \tag{21}$$

The concatenation of a set for feature maps generated by convolution increases the learning ability of latent features. Compared with ConvE, which captures the local relationships, ConvKB keeps the transitional characteristic and shows better experimental performance. HypER [55] utilizes hypernetwork $\mathbf{H}$ for 1D relation-specific convolutional filter generation to achieve multi-task knowledge sharing, and meanwhile simplifies 2D ConvE. It can also be interpreted as a tensor factorization model when taking hypernetwork and weight matrix as tensors.

5) Recurrent Neural Networks: The MLP- and CNN-based models, as mentioned above, learn triple-level representation. In comparison, the recurrent networks can capture long-term relational dependency in knowledge graphs. Gardner et al. [56] and Neelakantan et al. [57] propose RNN-based model over the relation path to learn vector representation without and with entity information, respectively. RSN [44] (Fig. 6d) designs a recurrent skip mechanism to enhance semantic representation learning by distinguishing relations and entities. The relational path as $(x_1, x_2, \ldots, x_T)$ with entities and relations in an alternating order is generated by random walk, and it is further used to calculate recurrent hidden state $\mathbf{h}_t = \tanh(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b})$. The skipping operation is conducted as

$$\mathbf{h}_t' = \begin{cases} \mathbf{h}_t & x_t \in \mathcal{E} \\ \mathbf{S}_1 \mathbf{h}_t + \mathbf{S}_2 \mathbf{x}_{t-1} & x_t \in \mathcal{R} \end{cases}, \tag{22}$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ are weight matrices.

6) Transformers: Transformer-based models have boosted contextualized text representation learning. To utilize contextual information in knowledge graphs, CoKE [45] employs transformers to encode edges and path sequences. Similarly, KG-BERT [58] borrows the idea form language model pre-training and takes Bidirectional Encoder Representations from Transformer (BERT) model as an encoder for entities and relations.

7) Graph Neural Networks: GNNs are introduced for learning connectivity structure under an encoder-decoder framework. R-GCN [59] proposes relation-specific transformation to model the directed nature of knowledge graphs. Its forward propagation is defined as

$$x_i^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} x_j^{(l)} + W_0^{(l)} x_i^{(l)}\right), \tag{23}$$

where $x_i^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of the i-th entity in l-th layer, N_i^r is a neighbor set of i-th entity within relation $r \in R$, $W_r^{(l)}$ and $W_0^{(l)}$ are the learnable parameter matrices, and $c_{i,r}$ is normalization such as $c_{i,r} = |N_i^r|$. Here, the GCN [60] acts as a graph encoder. To enable specific tasks, an encoder model still needs to be developed and integrated into the R-GCN framework. R-GCN takes the neighborhood of each entity equally. SACN [43] introduces weighted GCN (Fig. 6c), which defines the strength of two adjacent nodes with the same relation type, to capture the structural information in knowledge graphs by utilizing node structure, node attributes, and relation types. The decoder module called Conv-TransE adopts ConvE model as semantic matching metric and preserves the translational property. By aligning the convolutional outputs of entity and relation embeddings with C kernels to be $\mathbf{M}(\mathbf{h}, \mathbf{r}) \in \mathbb{R}^{C \times d}$, its scoring function is defined as

$$f_r(h, t) = g(\operatorname{vec}(\mathbf{M}(\mathbf{h}, \mathbf{r})) W) \mathbf{t}. \tag{24}$$

Nathani et al. [61] introduced graph attention networks with multi-head attention as encoder to capture multi-hop neighborhood features by inputing the concatenation of entity and relation embeddings. CompGCN [62] proposes entity-relation composition operations over each edge in the neighborhood of a central node and generalizes previous GCN-based models.

D. Embedding with Auxiliary Information

Multi-modal embedding incorporates external information such as text descriptions, type constraints, relational paths, and visual information, with a knowledge graph itself to facilitate more effective knowledge representation.

1) Textual Description: Entities in knowledge graphs have textual descriptions denoted as $\mathcal{D} = < w_1, w_2, \ldots, w_n >$, providing supplementary semantic information. The challenge of KRL with textual description is to embed both structured knowledge and unstructured textual information in the same space. Wang et al. [63] proposed two alignment models for aligning entity space and word space by introducing entity names and Wikipedia anchors. DKRL [64] extends TransE [15] to learn representation directly from entity descriptions by a convolutional encoder. SSP [65] captures the strong correlations between triples and textual descriptions by projecting them in a semantic subspace. The joint loss function is widely applied when incorporating KGE with textual description. Wang et al. [63] used a three-component loss $\mathcal{L} = \mathcal{L}_K + \mathcal{L}_T + \mathcal{L}_A$ of knowledge model $\mathcal{L}_K$, text model $\mathcal{L}_T$ and alignment model $\mathcal{L}_A$. SSP [65] uses a two-component objective function $\mathcal{L} = \mathcal{L}_{embed} + \mu \mathcal{L}_{topic}$ of embedding-specific loss $\mathcal{L}_{embed}$ and topic-specific loss $\mathcal{L}_{topic}$ within textual description, traded off by a parameter μ.

2) Type Information: Entities are represented with hierarchical classes or types, and consequently, relations with semantic types. SSE [66] incorporates semantic categories of entities to embed entities belonging to the same category smoothly in semantic space. TKRL [67] proposes type encoder model for projection matrix of entities to capture type hierarchy. Noticing that some relations indicate attributes of entities, KR-EAR [68] categorizes relation types into attributes and relations and modeled the correlations between entity descriptions. Zhang et al. [69] extended existing embedding methods with hierarchical relation structure of relation clusters, relations, and sub-relations.

3) Visual Information: Visual information (e.g., entity images) can be utilized to enrich KRL. Image-embodied IKRL [70], containing cross-modal structure-based and image-based representation, encodes images to entity space and follows the translation principle. The cross-modal representations make sure that structure-based and image-based representations are in the same representation space.

There remain many kinds of auxiliary information for KRL, such as attributes, relation paths, and logical rules. Wang et al. [5] gave a detailed review of these different kinds of information. This paper discusses relation path and logical rules under the umbrella of KGC in Sec. IV-A2 and IV-A4, respectively.

E. Summary

Knowledge representation learning is vital in the research community of knowledge graph. This section reviews four folds of KRL with several modern methods summarized in Table II and more in Appendix B. Overall, developing a novel KRL model is to answer the following four questions: 1) which representation space to choose; 2) how to measure the plausibility of triples in specific space; 3) what encoding model to modeling relational interaction; 4) whether to utilize auxiliary information.

The most popularly used representation space is Euclidean point-based space by embedding entities in vector space and modeling interactions via vector, matrix, or tensor. Other representation spaces, including complex vector space, Gaussian distribution, and manifold space and group, are also studied. Manifold space has an advantage over point-wise Euclidean space by relaxing the point-wise embedding. Gaussian embeddings can express the uncertainties of entities and relations, and multiple relation semantics. Embedding in complex vector space can effectively model different relational connectivity patterns, especially the symmetry/antisymmetry pattern. The representation space plays an essential role in encoding the semantic information of entities and capturing the relational properties. When developing a representation learning model, appropriate representation space should be selected and designed carefully to match the nature of encoding methods and balance the expressiveness and computational complexity. The scoring function with a distance-based metric utilizes the translation principle, while the semantic matching scoring function employs compositional operators. Encoding models, especially neural networks, play a critical role in modeling interactions of entities and relations. The bilinear models also have drawn much attention, and some tensor factorization can also be regarded as this family. Other methods incorporate auxiliary information of textual description, relation/entity types, and entity images.

TABLE II: A summary of recent KRL models. See more in Appendix B.

Model	Ent. & Rel. embed.	Scoring Function $f_r(h, t)$
RotatE [23]	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$, $\mathbf{r} \in \mathbb{C}^d$	$\lVert \mathbf{h} \circ \mathbf{r} - \mathbf{t} \rVert$
TorusE [30]	$[\mathbf{h}], [\mathbf{t}] \in \mathbb{T}^n$, $[\mathbf{r}] \in \mathbb{T}^n$	$\min_{(x,y)\in([h]+[r])\times[t]} \lVert x - y \rVert_i$
SimplE [47]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$, $\mathbf{r}, \mathbf{r}' \in \mathbb{R}^d$	$\frac{1}{2}\left(\mathbf{h} \circ \mathbf{r} \mathbf{t} + \mathbf{t} \circ \mathbf{r}' \mathbf{t}\right)$
TuckER [51]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}_e^d$, $\mathbf{r} \in \mathbb{R}_r^d$	$\mathcal{W} \times_1 \mathbf{h} \times_2 \mathbf{r} \times_3 \mathbf{t}$
ITransF [36]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\left\lVert \boldsymbol{\alpha}_r^H \cdot \mathbf{D} \cdot \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_r^T \cdot \mathbf{D} \cdot \mathbf{t} \right\rVert_\ell$
HolEx [40]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\sum_{j=0}^{l} p\left(\mathbf{h}, \boldsymbol{r}; \boldsymbol{c}_j\right) \cdot \boldsymbol{t}$
CrossE [41]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\sigma\left(\sigma\left(\mathbf{c}_r \circ \mathbf{h} + \mathbf{c}_r \circ \mathbf{h} \circ \mathbf{r} + \mathbf{b}\right) \mathbf{t}^\top\right)$
QuatE [24]	$\mathbf{h}, \mathbf{t} \in \mathbb{H}^d$, $\mathbf{r} \in \mathbb{H}^d$	$\mathbf{h} \otimes \frac{\mathbf{r}}{\vert \mathbf{r} \vert} \cdot \mathbf{t}$
SACN [43]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$g\left(\operatorname{vec}\left(\mathbf{M}\left(\mathbf{h}, \mathbf{r}\right)\right) W\right) \mathbf{t}$
ConvKB [42]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\operatorname{concat}\left(g\left(\left[\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}\right] * \omega\right)\right) \mathbf{w}$
ConvE [54]	$\mathbf{M}_h \in \mathbb{R}^{d_w \times d_h}, \mathbf{t} \in \mathbb{R}^d$ $\mathbf{M}_r \in \mathbb{R}^{d_w \times d_h}$	$\sigma\left(\operatorname{vec}\left(\sigma\left(\left[\mathbf{M}_h ; \mathbf{M}_r\right] * \boldsymbol{\omega}\right)\right) \mathbf{W}\right) \mathbf{t}$
DihEdral [31]	$\mathbf{h}^{(l)}, \mathbf{t}^{(l)} \in \mathbb{R}^2$ $\mathbf{R}^{(l)} \in \mathbb{D}_K$	$\sum_{l=1}^{L} \mathbf{h}^{(l)\top} \mathbf{R}^{(l)} \mathbf{t}^{(l)}$
HAKE [18]	$\mathbf{h}_m, \mathbf{t}_m \in \mathbb{R}^d, \mathbf{r}_m \in \mathbb{R}_+^d$ $\mathbf{h}_p, \mathbf{r}_p, \mathbf{t}_p \in [0, 2\pi)^d$	$- \lVert \mathbf{h}_m \circ \mathbf{r}_m - \mathbf{t}_m \rVert_2 -$ $\lambda \lVert \sin\left(\left(\mathbf{h}_p + \mathbf{r}_p - \mathbf{t}_p\right)/2\right) \rVert_1$
MuRP [28]	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{B}_c^d, b_h, b_t \in \mathbb{R}$	$-d_{\mathbb{B}}\left(\mathbf{h}^{(r)}, \mathbf{t}^{(r)}\right)^2 + b_s + b_o$
AttH [29]	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{B}_c^d, b_h, b_t \in \mathbb{R}$	$-d_{\mathbb{B}}^{c_r}\left(Q(h, r), \mathbf{e}_t^H\right)^2 + b_h + b_t$
LowFER [52]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d, \mathbf{r} \in \mathbb{R}^d$	$\left(\mathbf{S}^k \operatorname{diag}\left(\mathbf{U}^T \mathbf{h}\right) \mathbf{V}^T \mathbf{r}\right)^T \mathbf{t}$

IV. Knowledge Acquisition

Knowledge acquisition aims to construct knowledge graphs from unstructured text and other structured or semi-structured sources, complete an existing knowledge graph, and discover and recognize entities and relations. Well-constructed and large-scale knowledge graphs can be useful for many downstream applications and empower knowledge-aware models with commonsense reasoning, thereby paving the way for AI. The main tasks of knowledge acquisition include relation extraction, KGC, and other entity-oriented acquisition tasks such as entity recognition and entity alignment. Most methods formulate KGC and relation extraction separately. These two tasks, however, can also be integrated into a unified framework. Han et al. [71] proposed a joint learning framework with mutual attention for data fusion between knowledge graphs and text, which solves KGC and relation extraction from text. There are also other tasks related to knowledge acquisition, such as triple classification [72], relation classification [73], and open knowledge enrichment [74]. In this section, three-fold knowledge acquisition techniques on KGC, entity discovery, and relation extraction are reviewed thoroughly.

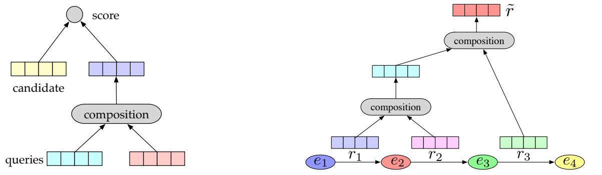
A. Knowledge Graph Completion

Because of the nature of incompleteness of knowledge graphs, KGC is developed to add new triples to a knowledge graph. Typical subtasks include link prediction, entity prediction, and relation prediction.

Preliminary research on KGC focused on learning low-dimensional embedding for triple prediction. In this survey, we term those methods as *embedding-based methods*. Most of them, however, failed to capture multi-step relationships. Thus, recent work turns to explore multi-step relation paths and incorporate logical rules, termed as *relation path inference* and *rule-based reasoning*, respectively. Triple classification as an associated task of KGC, which evaluates the correctness of a factual triple, is additionally reviewed in this section.

1) Embedding-based Models: Taking entity prediction as an example, embedding-based ranking methods, as shown in Fig. 7a, firstly learn embedding vectors based on existing triples. By replacing the tail entity or head entity with each entity $e \in \mathcal{E}$, those methods calculate scores of all the candidate entities and rank the top k entities. Aforementioned KRL methods (e.g., TransE [15], TransH [19], TransR [16], HolE [20], and R-GCN [59]) and joint learning methods like DKRL [64] with textual information can been used for KGC.

Unlike representing inputs and candidates in the unified embedding space, ProjE [75] proposes a combined embedding by space projection of the known parts of input triples, i.e., $(h, r, ?)$ or $(?, r, t)$, and the candidate entities with the candidate-entity matrix $\mathbf{W}^c \in \mathbb{R}^{s \times d}$, where s is the number of candidate entities. The embedding projection function including a neural combination layer and a output projection layer is defined as $h(\mathbf{e}, \mathbf{r}) = g\left(\mathbf{W}^c \sigma(\mathbf{e} \oplus \mathbf{r}) + b_p\right)$, where $\mathbf{e} \oplus \mathbf{r} = \mathbf{D}_e \mathbf{e} + \mathbf{D}_r \mathbf{r} + \mathbf{b}_c$ is the combination operator of input entity-relation pair. Previous embedding methods do not differentiate entities and relation prediction, and ProjE does not support relation prediction. Based on these observations, SENN [76] distinguishes three KGC subtasks explicitly by introducing a unified neural shared embedding with adaptively weighted general loss function to learn different latent features. Existing methods rely heavily on existing connections in knowledge graphs and fail to capture the evolution of factual knowledge or entities with a few connections. ConMask [77] proposes relationship-dependent content masking over the entity description to select relevant snippets of given relations, and CNN-based target fusion to complete the knowledge graph with unseen entities. It can only make a prediction when query relations and entities are explicitly expressed in the text description. Previous methods are discriminative models that rely on preprepared entity pairs or text corpus. Focusing on the medical domain, REMEDY [78] proposes a generative model, called conditional relationship variational autoencoder, for entity pair discovery from latent space.



(a) Embedding-based Ranking.

(b) Relation paths [57].

Fig. 7: Illustrations of embedding-based ranking and relation path reasoning.

2) Relation Path Reasoning: Embedding learning of entities and relations has gained remarkable performance in some benchmarks, but it fails to model complex relation paths. Relation path reasoning turns to leverage path information over the graph structure. Random walk inference has been widely investigated; for example, the Path-Ranking Algorithm (PRA) [79] chooses a relational path under a combination of path constraints and conducts maximum-likelihood classification. To improve path search, Gardner et al. [56] introduced vector space similarity heuristics in random work by incorporating textual content, which also relieves the feature sparsity issue in PRA. Neural multi-hop relational path modeling is also studied. Neelakantan et al. [57] developed an RNN model to compose the implications of relational paths by applying compositionality recursively (in Fig. 7b). Chain-of-Reasoning [80], a neural attention mechanism to enable multiple reasons, represents logical composition across all relations, entities, and text. Recently, DIVA [81] proposes a unified variational inference framework that takes multi-hop reasoning as two sub-steps of path-finding (a prior distribution for underlying path inference) and path-reasoning (a likelihood for link classification).

3) RL-based Path Finding: Deep reinforcement learning (RL) is introduced for multi-hop reasoning by formulating path-finding between entity pairs as sequential decision making, specifically a Markov decision process (MDP). The policy-based RL agent learns to find a step of relation to extending the reasoning paths via the interaction between the knowledge graph environment, where the policy gradient is utilized for training RL agents.

DeepPath [82] firstly applies RL into relational path learning and develops a novel reward function to improve accuracy, path diversity, and path efficiency. It encodes states in the continuous space via a translational embedding method and takes the relation space as its action space. Similarly, MINERVA [83] takes path walking to the correct answer entity as a sequential optimization problem by maximizing the expected reward. It excludes the target answer entity and provides more capable inference. Instead of using a binary reward function, Multi-Hop [84] proposes a soft reward mechanism. Action dropout is also adopted to mask some outgoing edges during training to enable more effective path exploration. M-Walk [85] applies an RNN controller to capture the historical trajectory and uses the Monte Carlo Tree Search (MCTS) for effective path generation. By leveraging text corpus with the sentence bag of current entity denoted as b_{e_t}, CPL [86] proposes collaborative policy learning for pathfinding and fact extraction from text.

With source, query and current entity denoted as e_s, e_q and e_t, and query relation denoted as r_q, the MDP environment and policy networks of these methods are summarized in Table III, where MINERVA, M-Walk and CPL use binary reward. For the policy networks, DeepPath uses fully-connected network, the extractor of CPL employs CNN, while the rest uses recurrent networks.

4) Rule-based Reasoning: To better make use of the symbolic nature of knowledge, another research direction of KGC is logical rule learning. A rule is defined by the head and body in the form of $head \leftarrow body$. The $head$ is an atom, i.e., a fact with variable subjects and/or objects, while the body can be a set of atoms. For example, given relations `sonOf`, `hasChild` and `gender`, and entities X and Y, there is a rule in the reverse form of logic programming as:

$$(Y, \texttt{sonOf}, X) \leftarrow (X, \texttt{hasChild}, Y) \wedge (Y, \texttt{gender}, Male)$$

Logical rules can been extracted by rule mining tools like AMIE [87]. The recent RLvLR [88] proposes a scalable rule mining approach with efficient rule searching and pruning, and uses the extracted rules for link prediction.

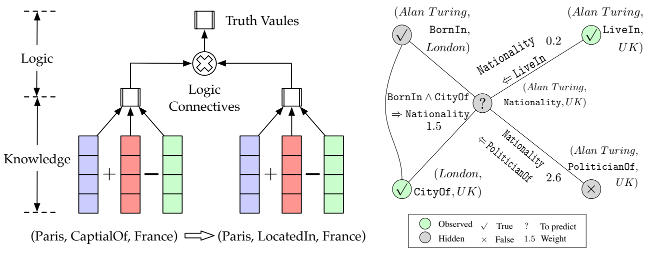
More research attention focuses on injecting logical rules into embeddings to improve reasoning, with joint learning or iterative training applied to incorporate first-order logic rules. For example, KALE [89] proposes a unified joint model with t-norm fuzzy logical connectives defined for compatible triples and logical rules embedding. Specifically, three compositions of logical conjunction, disjunction, and negation are defined to compose the truth value of a complex formula. Fig. 8a illustrates a simple first-order Horn clause inference. RUGE [90] proposes an iterative model, where soft rules are utilized for soft label prediction from unlabeled triples and labeled triples for embedding rectification. IterE [91] proposes an iterative training strategy with three components of embedding learning, axiom induction, and axiom injection.

The combination of neural and symbolic models has also attracted increasing attention to do rule-based reasoning in an end-to-end manner. Neural Theorem Provers (NTP) [92] learns logical rules for multi-hop reasoning, which utilizes a radial basis function kernel for differentiable computation on vector space. NeuralLP [93] enables gradient-based optimization to be applicable in the inductive logic programming, where a neural controller system is proposed by integrating attention mechanism and auxiliary memory. Neural-Num-LP [94] extends NeuralLP to learn numerical rules with dynamic programming and cumulative sum operations. pLogicNet [95] proposes probabilistic logic neural networks (Fig. 8b) to leverage first-order logic and learn effective embedding by combining the advantages of Markov logic networks and KRL methods while handling the uncertainty of logic rules. ExpressGNN [96] generalizes pLogicNet by tuning graph

TABLE III: Comparison of RL-based path finding for knowledge graph reasoning.

Method	State s_t	Action a_t	Reward γ	Policy Network
DeepPath [82]	$(\mathbf{e}_t, \mathbf{e}_q - \mathbf{e}_t)$	$\{r\}$	Global $1\ \mathrm{e_t} = \mathrm{e_q}$ or $-1\ \mathrm{e_t} \neq \mathrm{e_q}$ Efficiency $\frac{1}{\mathtt{length}(p)}$ Diversity $-\frac{1}{\lvert F\rvert} \sum_{i=1}^{\lvert F\rvert} \cos\left(\mathbf{p}, \mathbf{p}_i\right)$	Fully-connected network (FCN)
MINERVA [83]	(e_t, e_s, r_q, e_q)	$\{(e_t, r, v)\}$	$\mathbb{I}\left\{\mathrm{e_t} = \mathrm{e_q}\right\}$	$\mathbf{h}_t = LSTM\left(\mathbf{h_{t-1}}, [\mathbf{a_{t-1}}; \mathbf{o_t}]\right)$
Multi-Hop [84]	$(e_t, (e_s, r_q))$	$\left\{\left(r', e'\right) \mid \left(e_t, r', e'\right) \in \mathcal{G}\right\}$	$\gamma + (1-\gamma) f_{r_q}\left(e_s, e_T\right)$	$\mathbf{h}_t = LSTM\left(\mathbf{h}_{t-1}, \mathbf{a}_{t-1}\right)$
M-Walk [85]	$s_{t-1} \cup \left\{a_{t-1}, v_t, \mathcal{E}_{\mathcal{G}_{v_t}}, \mathcal{V}_{v_t}\right\}$	$\bigcup_t \mathcal{E}_{\mathcal{G}_{v_t}} \cup \{\mathrm{STOP}\}$	$\mathbb{I}\left\{\mathrm{e_t} = \mathrm{e_q}\right\}$	GRU-RNN + FCN
CPL [86] Reasoner	(e_s, r_q, h_t)	$\{\xi \in \mathcal{E}_{\mathcal{G}}\}$	$\mathbb{I}\left\{\mathrm{e_t} = \mathrm{e_q}\right\}$	$\mathbf{h}_t = LSTM\left(\mathbf{h}_{t-1}, [\mathbf{r}_t, \mathbf{e}_t]\right)$
CPL [86] Extractor	$\left(b_{e_t}, e_t\right)$	$\left\{\left(r', e'\right)\right\}_{\left(e_t, r', e'\right)} \in b_{e_t}$	step-wise delayed from reasoner	PCNN-ATT

networks and embedding and achieves more efficient logical reasoning.


Truth Vaules
Logic
Logic Connectives
Knowledge
(Paris, CaptialOf, France)
(Paris, LocatedIn, France)
(Alan Turing, BornIn, London)
(Alan Turing, LiveIn, UK)
Nationality ⇐ LiveIn
0.2
BornIn ∧ CityOf ⇒ Nationality
1.5
(Alan Turing, Nationality, UK)
?
Nationality ⇐ PoliticianOf
2.6
(Alan Turing, PoliticianOf, UK)
(London, CityOf, UK)
Observed
Hidden
✓ True
× False
? To predict
1.5 Weight

(a) KALE [89]. (b) pLogicNet [95].

Fig. 8: Illustrations of logical rule learning.

5) *Meta Relational Learning:* The long-tail phenomena exist in the relations of knowledge graphs. Meanwhile, the real-world scenario of knowledge is dynamic, where unseen triples are usually acquired. The new scenario, called as *meta relational learning* or *few-shot relational learning*, requires models to predict new relational facts with only a very few samples.

Targeting at the previous two observations, GMatching [97] develops a metric based few-shot learning method with entity embeddings and local graph structures. It encodes one-hop neighbors to capture the structural information with R-GCN and then takes the structural entity embedding for multi-step matching guided by long short-term memory (LSTM) networks to calculate the similarity scores. Meta-KGR [98], an optimization-based meta-learning approach, adopts model agnostic meta-learning for fast adaption and reinforcement learning for entity searching and path reasoning. Inspired by model-based and optimization-based meta-learning, MetaR [99] transfers relation-specific meta information from support set to query set, and archives fast adaption via loss gradient of high-order relational representation. Zhang et al. [100] proposed joint modules of heterogeneous graph encoder, recurrent autoencoder, and matching network to complete new relational facts with few-shot references. Qin et al. [101] utilized GAN to generate reasonable embeddings for unseen relations under the zero-shot learning setting.

6) *Triple Classification:* Triple classification is to determine whether facts are correct in testing data, which is typically regarded as a binary classification problem. The decision rule is based on the scoring function with a specific threshold. Aforementioned embedding methods could be applied for triple classification, including translational distance-based methods like TransH [19] and TransR [16] and semantic matching-based methods such as NTN [17], HolE [20] and ANALOGY [21].

Vanilla vector-based embedding methods failed to deal with 1-to-n relations. Recently, Dong et al. [72] extended the embedding space into region-based n-dimensional balls where the tail region is in the head region for 1-to-n relation using fine-grained type chains, i.e., tree-structure conceptual clusterings. This relaxation of embedding to n-balls turns triple classification into a geometric containment problem and improves the performance for entities with long type chains. However, it relies on the type chains of entities and suffers from the scalability problem.

B. Entity Discovery

This section distinguishes entity-based knowledge acquisition into several fractionized tasks, i.e., entity recognition, entity disambiguation, entity typing, and entity alignment. We term them as *entity discovery* as they all explore entity-related knowledge under different settings.

1) *Entity Recognition:* Entity recognition or named entity recognition (NER), when it focuses on specifically named entities, is a task that tags entities in text. Hand-crafted features such as capitalization patterns and language-specific resources like gazetteers are applied in many pieces of literature. Recent work applies sequence-to-sequence neural architectures, for example, LSTM-CNN [102] for learning character-level and word-level features and encoding partial lexicon matches. Lample et al. [103] proposed stacked neural architectures by stacking LSTM layers and CRF layers, i.e., LSTM-CRF (in Fig. 9a) and Stack-LSTM. MGNER [104] proposes an integrated framework with entity position detection in various granularities and attention-based entity classification for both nested and non-overlapping named entities. Hu et al. [105] distinguished multi-token and single-token entities with multi-task training. Recently, Li et al. [106] formulated flat and nested NER as a unified machine reading comprehension framework by referring annotation guidelines to construct query questions.

2) *Entity Typing:* Entity typing includes coarse and fine-grained types, while the latter uses a tree-structured type category and is typically regarded as multi-class and multi-label classification. To reduce label noise, PLE [107] focuses on correct type identification and proposes a partial-label embedding model with a heterogeneous graph for the representation of entity mentions, text features, and entity types and their relationships. To tackle the increasing growth of typeset and

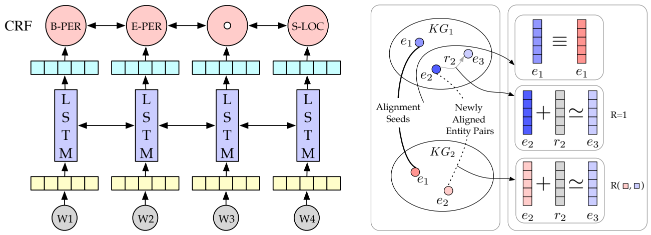
noisy labels, Ma et al. [108] proposed prototype-driven label embedding with hierarchical information for zero-shot fine-grained named entity typing.

3) Entity Disambiguation: Entity disambiguation or entity linking is a unified task which links entity mentions to the corresponding entities in a knowledge graph. For example, Einstein won the Noble Prize in Physics in 1921. The entity mention of "Einstein" should be linked to the entity of `Albert Einstein`. The contemporary end-to-end learning approaches have made efforts through representation learning of entities and mentions, for example, DSRM [109] for modeling entity semantic relatedness and EDKate [110] for the joint embedding of entity and text. Ganea and Hofmann [111] proposed an attentive neural model over local context windows for entity embedding learning and differentiable message passing for inferring ambiguous entities. By regarding relations between entities as latent variables, Le and Titov [112] developed an end-to-end neural architecture with relation-wise and mention-wise normalization.

4) Entity Alignment: The tasks, as mentioned earlier, involve entity discovery from text or a single knowledge graph, while entity alignment (EA) aims to fuse knowledge among various knowledge graphs. Given $\mathcal{E}_1$ and $\mathcal{E}_2$ as two different entity sets of two different knowledge graphs, EA is to find an alignment set $A = \{(e_1, e_2) \in \mathcal{E}_1 \times \mathcal{E}_2 | e_1 \equiv e_2\}$, where entity e_1 and entity e_2 hold an equivalence relation $\equiv$. In practice, a small set of alignment seeds (i.e., synonymous entities appear in different knowledge graphs) is given to start the alignment process, as shown in the left box of Fig. 9b.

Embedding-based alignment calculates the similarity between the embeddings of a pair of entities. IPTransE [113] maps entities into a unified representation space under a joint embedding framework (Fig. 9b) through aligned translation as $\left\|\mathbf{e}_1 + \mathbf{r}^{(\mathcal{E}_1 \to \mathcal{E}_2)} - \mathbf{e}_2\right\|$, linear transformation as $\left\|\mathbf{M}^{(\mathcal{E}_1 \to \mathcal{E}_2)}\mathbf{e}_1 - \mathbf{e}_2\right\|$, and parameter sharing as $\mathbf{e}_1 \equiv \mathbf{e}_2$. To solve error accumulation in iterative alignment, BootEA [114] proposes a bootstrapping approach in an incremental training manner, together with an editing technique for checking newly-labeled alignment.

Additional information of entities is also incorporated for refinement, for example, JAPE [115] capturing the correlation between cross-lingual attributes, KDCoE [116] embedding multi-lingual entity descriptions via co-training, MultiKE [117] learning multiple views of the entity name, relation, and attributes, and alignment with character attribute embedding [118].



(a) Entity recognition with LSTM-CRF [103].

(b) Entity alignment with IP-TransE [113].

Fig. 9: Illustrations of several entity discovery tasks.

C. Relation Extraction

Relation extraction is a key task to build large-scale knowledge graphs automatically by extracting unknown relational facts from plain text and adding them into knowledge graphs. Due to the lack of labeled relational data, distant supervision [119], also referred to as weak supervision or self-supervision, uses heuristic matching to create training data by assuming that sentences containing the same entity mentions may express the same relation under the supervision of a relational database. Mintz et al. [120] adopted the distant supervision for relation classification with textual features, including lexical and syntactic features, named entity tags, and conjunctive features. Traditional methods rely highly on feature engineering [120], with a recent approach exploring the inner correlation between features [121]. Deep neural networks are changing the representation learning of knowledge graphs and texts. This section reviews recent advances of neural relation extraction (NRE) methods, with an overview illustrated in Fig. 10.

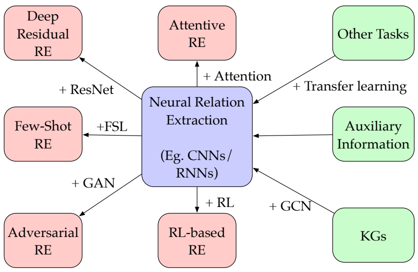


Fig. 10: An overview of neural relation extraction.

1) Neural Relation Extraction: Trendy neural networks are widely applied to NRE. CNNs with position features of relative distances to entities [122] are firstly explored for relation classification, and then extended to relation extraction by multi-window CNN [123] with multiple sized convolutional filters. Multi-instance learning takes a bag of sentences as input to predict the relationship of the entity pair. PCNN [124] applies the piecewise max pooling over the segments of convolutional representation divided by entity position. Compared with vanilla CNN [122], PCNN can more efficiently capture the structural information within the entity pair. MIMLCNN [125] further extends it to multi-label learning with cross-sentence max pooling for feature selection. Side information such as class ties [126] and relation path [127] is also utilized.

RNNs are also introduced, for example, SDP-LSTM [128] adopts multi-channel LSTM while utilizing the shortest dependency path between entity pair, and Miwa et al. [129] stacks sequential and tree-structure LSTMs based on dependency tree. BRCNN [130] combines RNN for capturing sequential dependency with CNN for representing local semantics using two-channel bidirectional LSTM and CNN.

2) Attention Mechanism: Many variants of attention mechanisms are combined with CNNs, for example, word-level attention to capturing semantic information of words [131] and selective attention over multiple instances to alleviate the impact of noisy instances [132]. Other side information is also introduced for enriching semantic representation. APCNN [133] introduces entity description by PCNN and sentence-level attention, while HATT [134] proposes hierarchical selective attention to capture the relation hierarchy by concatenating attentive representation of each hierarchical layer. Rather than CNN-based sentence encoders, Att-BLSTM [73] proposes word-level attention with BiLSTM.

3) Graph Convolutional Networks: GCNs are utilized for encoding dependency tree over sentences or learning KGEs to leverage relational knowledge for sentence encoding. C-GCN [135] is a contextualized GCN model over the pruned dependency tree of sentences after path-centric pruning. AG-GCN [136] also applies GCN over the dependency tree, but utilizes multi-head attention for edge selection in a soft weighting manner. Unlike previous two GCN-based models, Zhang et al., [137] applied GCN for relation embedding in knowledge graph for sentence-based relation extraction. The authors further proposed a coarse-to-fine knowledge-aware attention mechanism for the selection of informative instance.

4) Adversarial Training: Adversarial Training (AT) is applied to add adversarial noise to word embeddings for CNN- and RNN-based relation extraction under the MIML learning setting [138]. DSGAN [139] denoises distantly supervised relation extraction by learning a generator of sentence-level true positive samples and a discriminator that minimizes the probability of being true positive of the generator.

5) Reinforcement Learning: RL has been integrated into neural relation extraction recently by training instance selector with policy network. Qin et al. [140] proposed to train policy-based RL agent of sentential relation classifier to redistribute false positive instances into negative samples to mitigate the effect of noisy data. The authors took the F1 score as an evaluation metric and used F1 score based performance change as the reward for policy networks. Similarly, Zeng et al. [141] and Feng et al. [142] proposed different reward strategies. The advantage of RL-based NRE is that the relation extractor is model-agnostic. Thus, it could be easily adapted to any neural architectures for effective relation extraction. Recently, HRL [143] proposed a hierarchical policy learning framework of high-level relation detection and low-level entity extraction.

6) Other Advances: Other advances of deep learning are also applied for neural relation extraction. Noticing that current NRE methods do not use very deep networks, Huang and Wang [144] applied deep residual learning to noisy relation extraction and found that 9-layer CNNs have improved performance. Liu et al. [145] proposed to initialize the neural model by transfer learning from entity classification. The cooperative CORD [146] ensembles text corpus and knowledge graph with external logical rules by bidirectional knowledge distillation and adaptive imitation. TK-MF [147] enriches sentence representation learning by matching sentences and topic words. Recently, Shahbazi et al. [148] studied trustworthy relation extraction by benchmarking several explanation mechanisms, including saliency, gradient × input, and leave one out.

The existence of low-frequency relations in knowledge graphs requires few-shot relation classification with unseen classes or only a few instances. Gao et al. [149] proposed hybrid attention-based prototypical networks to compute prototypical relation embedding and compare its distance between the query embedding. Qin et al. [150] explored the relationships between relations with a global relation graph and formulated few-shot relation extraction as a Bayesian meta-learning problem to learn the posterior distribution of relations' prototype vectors.

D. Summary

This section reviews knowledge completion for incomplete knowledge graph and acquisition from plain text.

Knowledge graph completion completes missing links between existing entities or infers entities given entity and relation queries. Embedding-based KGC methods generally rely on triple representation learning to capture semantics and do candidate ranking for completion. Embedding-based reasoning remains in individual relation level, and is poor at complex reasoning because it ignores the symbolical nature of knowledge graph, and lack of interpretability. Hybrid methods with symbolics and embedding incorporate rule-based reasoning, overcome the sparsity of knowledge graph to improve the quality of embedding, facilitate efficient rule injection, and induce interpretable rules. With the observation of the graphical nature of knowledge graphs, path search and neural path representation learning are studied. However, they suffer from connectivity deficiency when traverses over large-scale graphs. The emerging direction of meta relational learning aims to learn fast adaptation over unseen relations in low-resource settings.

Entity discovery acquires entity-oriented knowledge from text and fuses knowledge between knowledge graphs. There are several categories according to specific settings. Entity recognition is explored in a sequence-to-sequence manner, entity typing discusses noisy type labels and zero-shot typing, and entity disambiguation and alignment learn unified embeddings with iterative alignment model proposed to tackle the issue of a limited number of alignment seed. However, it may face error accumulation problems if newly-aligned entities suffer from poor performance. Language-specific knowledge has increased in recent years and consequentially motivates the research on cross-lingual knowledge alignment.

Relation extraction suffers from noisy patterns under the assumption of distant supervision, especially in text corpus of different domains. Thus, weakly supervised relation extraction must mitigate the impact of noisy labeling. For example, multi-instance learning takes bags of sentences as inputs and attention mechanism [132] reduce noisy patterns by soft selection over instances, and RL-based methods formulate instance selection as a hard decision. Another principle is to learn richer representation as possible. As deep neural networks can solve error propagation in traditional feature extraction methods, this field is dominated by DNN-based models, as summarized in Table IV.

V. Temporal Knowledge Graph

Current knowledge graph research mostly focuses on static knowledge graphs where facts are not changed with time, while

TABLE IV: A summary of neural relation extraction and recent advances.

Category	Method	Mechanism	Auxiliary Information
CNNs	O-CNN [122]	CNN + max pooling	position embedding
	Multi CNN [123]	Multi-window convolution + max pooling	position embedding
	PCNN [124]	CNN + piecewise max pooling	position embedding
	MIMLCNN [125]	CNN + piecewise and cross-sentence max pooling	position embedding
	Ye et al. [126]	CNN/PCNN + pairwise ranking	position embedding, class ties
	Zeng et al. [127]	CNN + max pooling	position embedding, relation path
RNNs	SDP-LSTM [128]	Multichannel LSTM + dropout	dependency tree, POS, GR, hypernyms
	LSTM-RNN [129]	Bi-LSTM + Bi-TreeLSTM	POS, dependency tree
	BRCNN [130]	Two-channel LSTM + CNN + max pooling	dependency tree, POS, NER
Attention	Attention-CNN [131]	CNN + word-level attention + max pooling	POS, position embedding
	Lin et al. [132]	CNN/PCNN + selective attention + max pooling	position embedding
	Att-BLSTM [73]	Bi-LSTM + word-level attention	position indicator
	APCNN [133]	PCNN + sentence-level attention	entity descriptions
	HATT [134]	CNN/PCNN + hierarchical attention	position embedding, relation hierarchy
GCNs	C-GCN [135]	LSTM + GCN + path-centric pruning	dependency tree
	KATT [137]	Pre-training + GCN + CNN + attention	position embedding, relation hierarchy
	AGGCN [136]	GCN + multi-head attention + dense layers	dependency tree
Adversarial	Wu et al. [138]	AT + PCNN/RNN + selective attention	indicator encoding
	DSGAN [139]	GAN + PCNN/CNN + attention	position embedding
RL	Qin et al. [140]	Policy gradient + CNN + performance change reward	position embedding
	Zeng et al. [141]	Policy gradient + CNN + +1/-1 bag-result reward	position embedding
	Feng et al. [142]	Policy gradient + CNN + predictive probability reward	position embedding
	HRL [143]	Hierarchical policy learning + Bi-LSTM + MLP	relation indicator
Others	ResCNN-x [144]	Residual convolution block + max pooling	position embedding
	Liu et al. [145]	Transfer learning + sub-tree parse + attention	position embedding
	CORD [146]	BiGRU + hierarchical attention + cooperative module	position embedding, logic rules
	TK-MF [147]	Topic modeling + multi-head self attention	position embedding, topic words
	HATT-Proto [149]	Prototypical networks + CNN + hybrid attention	position embedding
	REGRAB [150]	Stochastic gradient Langevin dynamics + BERT + GNN	global relation graph

the temporal dynamics of a knowledge graph is less explored. However, the temporal information is of great importance because the structured knowledge only holds within a specific period, and the evolution of facts follows a time sequence. Recent research begins to take temporal information into KRL and KGC, which is termed as *temporal knowledge graph* in contrast to the previous static knowledge graph. Research efforts have been made for learning temporal and relational embedding simultaneously.

A. Temporal Information Embedding

Temporal information is considered in temporal-aware embedding by extending triples into temporal quadruple as (h, r, t, τ), where τ provides additional temporal information about when the fact held. Leblay and Chekol [151] investigated temporal scope prediction over time-annotated triple, and simply extended existing embedding methods, for example, TransE with the vector-based TTransE defined as

$$f_\tau(h, r, t) = -\|\mathbf{h} + \mathbf{r} + \tau - \mathbf{t}\|_{L_{1/2}}. \tag{25}$$

Ma et al. [152] also generalized existing static embedding methods and proposed ConT by replacing the shared weight vector of Tucker with a timestamp embedding. Temporally scoped quadruple extends triples by adding a time scope $[\tau_s, \tau_e]$, where τ_s and τ_e stand for the beginning and ending of the valid period of a triple, and then a static subgraph G_τ can be derived from the dynamic knowledge graph when given a specific timestamp τ. HyTE [153] takes a time stamp as a hyperplane $\mathbf{w}_\tau$ and projects entity and relation representation as $P_\tau(\mathbf{h}) = \mathbf{h} - \left(\mathbf{w}_\tau^\top \mathbf{h}\right)\mathbf{w}_\tau$, $P_\tau(\mathbf{t}) = \mathbf{t} - \left(\mathbf{w}_\tau^\top \mathbf{t}\right)\mathbf{w}_\tau$, and $P_\tau(\mathbf{r}) = \mathbf{r} - \left(\mathbf{w}_\tau^\top \mathbf{r}\right)\mathbf{w}_\tau$. The temporally projected scoring function is calculated as

$$f_\tau(h, r, t) = \|P_\tau(\mathbf{h}) + P_\tau(\mathbf{r}) - P_\tau(\mathbf{t})\|_{L_1/L_2} \tag{26}$$

within the projected translation of $P_\tau(\mathbf{h}) + P_\tau(\mathbf{r}) \approx P_\tau(\mathbf{t})$. García-Durán et al. [154] concatenated predicate token sequence and temporal token sequence, and used LSTM to encode the concatenated time-aware predicate sequences. The last hidden state of LSTM is taken as temporal-aware relational embedding r_{temp}. The scoring function of extended TransE and DistMult are calculated as $\|\mathbf{h} + \mathbf{r}_{temp} - \mathbf{t}\|_2$ and $(\mathbf{h} \circ \mathbf{t})\,\mathbf{r}_{temp}^T$, respectively. By defining the context of an entity e as an aggregate set of facts containing e, Liu et al. [155] proposed context selection to capture useful contexts, and measured temporal consistency with selected context. By formulating temporal KGC as 4-order tensor completion, Lacroix et al. [156] proposed TComplEx, which extends ComplEx decomposition, and introduced weighted regularizers.

B. Entity Dynamics

Real-world events change entities' state, and consequently, affect the corresponding relations. To improve temporal scope inference, the contextual temporal profile model [157] formulates the temporal scoping problem as state change detection and utilizes the context to learn state and state change vectors. Inspired by the diachronic word embedding, Goel et al. [158] took an entity and timestamp as the input of entity embedding function to preserve the temporal-aware characteristics of entities at any time point. Know-evolve [159], a deep evolutionary knowledge network, investigates the knowledge evolution phenomenon of entities and their evolved relations. A multivariate temporal point process is used to model the occurrence of facts, and a novel recurrent network is developed to learn the representation of non-linear temporal evolution. To capture the interaction between nodes, RE-NET [160] models event sequences via RNN-based event encoder, and neighborhood aggregator. Specifically, RNN is used to capture the temporal entity interaction, and the neighborhood aggregator aggregates the concurrent interactions.

C. Temporal Relational Dependency

There exists temporal dependencies in relational chains following the timeline, for example, `wasBornIn` → `graduateFrom` → `workAt` → `diedIn`. Jiang et al. [161], [162] proposed time-aware embedding, a joint learning framework with temporal regularization, to incorporate temporal order and consistency information. The authors defined a temporal scoring function as

$$f\left(\langle r_k, r_l\rangle\right) = \left\|\mathbf{r}_k\mathbf{T} - \mathbf{r}_l\right\|_{L_{1/2}}, \tag{27}$$

where $\mathbf{T} \in \mathbb{R}^{d\times d}$ is an asymmetric matrix that encodes the temporal order of relation, for a temporal ordering relation pair $\langle r_k, r_l\rangle$. Three temporal consistency constraints of disjointness, ordering, and spans are further applied by integer linear programming formulation.

D. Temporal Logical Reasoning

Logical rules are also studied for temporal reasoning. Chekol et al. [163] explored Markov logic network and probabilistic soft logic for reasoning over uncertain temporal knowledge graphs. RLvLR-Stream [88] considers temporal close-path rules and learns the structure of rules from the knowledge graph stream for reasoning.

VI. Knowledge-Aware Applications

Rich structured knowledge can be useful for AI applications. However, how to integrate such symbolic knowledge into the computational framework of real-world applications remains a challenge. This section introduces several recent DNN-based knowledge-driven approaches with the applications on NLU, recommendation, and question answering. More miscellaneous applications such as digital health and search engine are introduced in Appendix D.

A. Natural Language Understanding

Knowledge-aware NLU enhances language representation with structured knowledge injected into a unified semantic space. Recent knowledge-driven advances utilize explicit factual knowledge and implicit language representation, with many NLU tasks explored. Chen et al. [164] proposed double-graph random walks over two knowledge graphs, i.e., a slot-based semantic knowledge graph and a word-based lexical knowledge graph, to consider inter-slot relations in spoken language understanding. Wang et al. [165] augmented short text representation learning with knowledge-based conceptualization by a weighted word-concept embedding. Peng et al. [166] integrated an external knowledge base to build a heterogeneous information graph for event categorization in short social text.

Language modeling as a fundamental NLP task predicts the next word given preceding words in the given sequence. Traditional language modeling does not exploit factual knowledge with entities frequently observed in the text corpus. How to integrate knowledge into language representation has drawn increasing attention. Knowledge graph language model (KGLM) [167] learns to render knowledge by selecting and copying entities. ERNIE-Tsinghua [168] fuses informative entities via aggregated pre-training and random masking. BERT-MK [169] encodes graph contextualized knowledge and focuses on the medical corpus. K-BERT [170] infuses domain knowledge into BERT contextual encoder. ERNIE-Baidu [171] introduces named entity masking and phrase masking to integrate knowledge into the language model and is further improved by ERNIE 2.0 [172] via continual multi-task learning. Rethinking about large-scale training on language model and querying over knowledge graphs, Petroni et al. [173] analyzed the language model and knowledge base. They found that certain factual knowledge can be acquired via pre-training language model.

B. Question Answering

knowledge-graph-based question answering (KG-QA) answers natural language questions with facts from knowledge graphs. Neural network-based approaches represent questions and answers in distributed semantic space, and some also conduct symbolic knowledge injection for commonsense reasoning.

1) Single-fact QA: Taking a knowledge graph as an external intellectual source, simple factoid QA or single-fact QA is to answer a simple question involving a single knowledge graph fact. Bordes et al. [174] adapted memory network for simple question answering, taking knowledge base as external memory. Dai et al. [175] proposed a conditional focused neural network equipped with focused pruning to reduce the search space. To generate natural answers in a user-friendly way, COREAQ [176] introduces copying and retrieving mechanisms to generate smooth and natural responses in a seq2seq manner, where an answer is predicted from the corpus vocabulary, copied from the given question or retrieved from the knowledge graph. BAMnet [177] models the two-way interaction between questions and knowledge graph with a bidirectional attention mechanism.

Although deep learning techniques are intensively applied in KG-QA, they inevitably increase the model complexity. Through the evaluation of simple KG-QA with and without neural networks, Mohammed et al. [178] found that sophisticated deep models such as LSTM and gated recurrent unit (GRU) with heuristics achieve state of the art, and non-neural models also gain reasonably well performance.

2) Multi-hop Reasoning: Those neural network-based methods gain improvements with the combination of neural encoder-decoder models, but to deal with complex multi-hop relation requires a more dedicated design to be capable of multi-hop commonsense reasoning. Structured knowledge provides informative commonsense observations and acts as relational inductive biases, which boosts recent studies on commonsense knowledge fusion between symbolic and semantic space for multi-hop reasoning. Bauer et al. [179] proposed multi-hop bidirectional attention and pointer-generator decoder for effective multi-hop reasoning and coherent answer generation, utilizing external commonsense knowledge by relational path selection from ConceptNet and injection with selectively-gated attention. Variational Reasoning Network (VRN) [180] conducts multi-hop logic reasoning with reasoning-graph embedding, while handles the uncertainty in topic entity recognition. KagNet [181] performs concept recognition to build a schema graph from ConceptNet and learns path-based relational representation via GCN, LSTM, and hierarchical path-based attention. CogQA [182] combines implicit extraction and explicit reasoning and proposes a cognitive graph model based on BERT and GNN for multi-hop QA.

C. Recommender Systems

Recommender systems have been widely explored by collaborative filtering, which makes use of users' historical information. However, it often fails to solve the sparsity issue and the cold start problem. Integrating knowledge graphs as external information enables recommendation systems to have the ability of commonsense reasoning.

By injecting knowledge-graph-based side information such as entities, relations, and attributes, many efforts work on embedding-based regularization to improve recommendation. The collaborative CKE [183] jointly trains KGEs, item's textual information, and visual content via translational KGE model and stacked auto-encoders. Noticing that time-sensitive and topic-sensitive news articles consist of condensed entities and common knowledge, DKN [184] incorporates knowledge graph by a knowledge-aware CNN model with multi-channel word-entity-aligned textual inputs. However, DKN cannot be trained in an end-to-end manner as it needs to learn entity embedding in advance. To enable end-to-end training, MKR [185] associates multi-task knowledge graph representation and recommendation by sharing latent features and modeling high-order item-entity interaction. While other works consider the relational path and structure of knowledge graphs, KPRN [186] regards the interaction between users and items as an entity-relation path in the knowledge graph and conducts preference inference over the path with LSTM to capture the sequential dependency. PGPR [187] performs reinforcement policy-guided path reasoning over knowledge-graph-based user-item interaction. KGAT [188] applies graph attention network over the collaborative knowledge graph of entity-relation and user-item graphs to encode high-order connectivities via embedding propagation and attention-based aggregation.

VII. Future Directions

Many efforts have been conducted to tackle the challenges of knowledge representation and its related applications. However, there remains several formidable open problems and promising future directions.

A. Complex Reasoning

Numerical computing for knowledge representation and reasoning requires a continuous vector space to capture the semantic of entities and relations. While embedding-based methods have a limitation on complex logical reasoning, two directions on the relational path and symbolic logic are worthy of being further explored. Some promising methods such as recurrent relational path encoding, GNN-based message passing over knowledge graph, and reinforcement learning-based pathfinding and reasoning are up-and-coming for handling complex reasoning. For the combination of logic rules and embeddings, recent works [95], [96] combine Markov logic networks with KGE, aiming to leverage logic rules and handling their uncertainty. Enabling probabilistic inference for capturing the uncertainty and domain knowledge with efficiently embedding will be a noteworthy research direction.

B. Unified Framework

Several representation learning models on knowledge graphs have been verified as equivalence, for example, Hayshi and Shimbo [189] proved that HolE and ComplEx are mathematically equivalent for link prediction with a particular constraint. ANALOGY [21] provides a unified view of several representative models, including DistMult, ComplEx, and HolE. Wang et al. [46] explored connections among several bilinear models. Chandrahas et al. [190] explored the geometric understanding of additive and multiplicative KRL models. Most works formulated knowledge acquisition KGC and relation extraction separately with different models. Han et al. [71] put them under the same roof and proposed a joint learning framework with mutual attention for information sharing between knowledge graph and text. A unified understanding of knowledge representation and reasoning is less explored. An investigation towards unification in a way similar to the unified framework of graph networks [191], however, will be worthy of bridging the research gap.

C. Interpretability

Interpretability of knowledge representation and injection is a vital issue for knowledge acquisition and real-world applications. Preliminary efforts have been made for interpretability. ITransF [36] uses sparse vectors for knowledge transferring and interprets with attention visualization. CrossE [41] explores the explanation scheme of knowledge graphs by using embedding-based path searching to generate explanations for link prediction. However, recent neural models have limitations on transparency and interpretability, although they have gained impressive performance. Some methods combine black-box neural models and symbolic reasoning by incorporating logical rules to increase the interoperability. Interpretability can convince people to trust predictions. Thus, further work should go into interpretability and improve the reliability of predicted knowledge.

D. Scalability

Scalability is crucial in large-scale knowledge graphs. There is a trade-off between computational efficiency and model expressiveness, with a limited number of works applied to more than 1 million entities. Several embedding methods use simplification to reduce the computation cost, such as simplifying tensor products with circular correlation operation [20].

However, these methods still struggle with scaling to millions of entities and relations.

Probabilistic logic inference using Markov logic networks is computationally intensive, making it hard to scalable to large-scale knowledge graphs. Rules in a recent neural logical model [95] are generated by simple brute-force search, making it insufficient on large-scale knowledge graphs. ExpressGNN [96] attempts to use NeuralLP [93] for efficient rule induction. Nevertheless, there still has a long way to go to deal with cumbersome deep architectures and the increasingly growing knowledge graphs.

E. Knowledge Aggregation

The aggregation of global knowledge is the core of knowledge-aware applications. For example, recommendation systems use a knowledge graph to model user-item interaction and text classification jointly to encode text and knowledge graph into a semantic space. Most current knowledge aggregation methods design neural architectures such as attention mechanisms and GNNs. The natural language processing community has been boosted from large-scale pre-training via transformers and variants like BERT models. At the same time, a recent finding [173] reveals that the pre-training language model on the unstructured text can acquire certain factual knowledge. Large-scale pre-training can be a straightforward way to injecting knowledge. However, rethinking the way of knowledge aggregation in an efficient and interpretable manner is also of significance.

F. Automatic Construction and Dynamics

Current knowledge graphs rely highly on manual construction, which is labor-intensive and expensive. The widespread applications of knowledge graphs on different cognitive intelligence fields require automatic knowledge graph construction from large-scale unstructured content. Recent research mainly works on semi-automatic construction under the supervision of existing knowledge graphs. Facing the multimodality, heterogeneity, and large-scale application, automatic construction is still of great challenge.

The mainstream research focuses on static knowledge graphs, with several works on predicting temporal scope validity and learning temporal information and entity dynamics. Many facts only hold within a specific period. A dynamic knowledge graph, together with learning algorithms capturing dynamics, can address the limitation of traditional knowledge representation and reasoning by considering the temporal nature.

VIII. Conclusion

Knowledge graphs as the ensemble of human knowledge have attracted increasing research attention, with the recent emergence of knowledge representation learning, knowledge acquisition methods, and a wide variety of knowledge-aware applications. The paper conducts a comprehensive survey on the following four scopes: 1) knowledge graph embedding, with a full-scale systematic review from embedding space, scoring metrics, encoding models, embedding with external information, and training strategies; 2) knowledge acquisition of entity discovery, relation extraction, and graph completion from three perspectives of embedding learning, relational path inference and logical rule reasoning; 3) temporal knowledge graph representation learning and completion; 4) real-world knowledge-aware applications on natural language understanding, recommendation systems, question answering and other miscellaneous applications. Besides, some useful resources of datasets and open-source libraries, and future research directions are introduced and discussed. Knowledge graph hosts a large research community and has a wide range of methodologies and applications. We conduct this survey to have a summary of current representative research efforts and trends and expect it can facilitate future research.

Appendix A
Mathematical Operations

Hermitian dot product (Eq. 28) and Hamilton product (Eq. 29) are used in complex vector space (Sec. III-A2). Given $\mathbf{h}$ and $\mathbf{t}$ represented in complex space $\mathbb{C}^d$, the Hermitian dot product $\langle , \rangle : \mathbb{C}^d \times \mathbb{C}^d \longrightarrow \mathbb{C}$ is calculated as the sesquilinear form of

$$\langle \mathbf{h}, \mathbf{t} \rangle = \overline{\mathbf{h}}^T \mathbf{t}, \tag{28}$$

where $\overline{\mathbf{h}} = \mathrm{Re}(\mathbf{h}) - i\,\mathrm{Im}(\mathbf{h})$ is the conjugate operation over $\mathbf{h} \in \mathbb{C}^d$. The quaternion extends complex numbers into four-dimensional hypercomplex space. With two d-dimensional quaternions defined as $\mathbf{Q_1} = \mathbf{a_1} + \mathbf{b_1}\mathbf{i} + \mathbf{c_1}\mathbf{j} + \mathbf{d_1}\mathbf{k}$ and $\mathbf{Q_2} = \mathbf{a_2} + \mathbf{b_2}\mathbf{i} + \mathbf{c_2}\mathbf{j} + \mathbf{d_2}\mathbf{k}$, the Hamilton product $\otimes : \mathbb{H}^d \times \mathbb{H}^d \to \mathbb{H}^d$ is defined as

$$\begin{aligned}\mathbf{Q_1} \otimes \mathbf{Q_2} = &\left(\mathbf{a_1} \circ \mathbf{a_2} - \mathbf{b_1} \circ \mathbf{b_2} - \mathbf{c_1} \circ \mathbf{c_2} - \mathbf{d_1} \circ \mathbf{d_2}\right) \\ &+ \left(\mathbf{a_1} \circ \mathbf{b_2} + \mathbf{b_1} \circ \mathbf{a_2} + \mathbf{c_1} \circ \mathbf{d_2} - \mathbf{d_1} \circ \mathbf{c_2}\right) \mathbf{i} \\ &+ \left(\mathbf{a_1} \circ \mathbf{c_2} - \mathbf{b_1} \circ \mathbf{d_2} + \mathbf{c_1} \circ \mathbf{a_2} + \mathbf{d_1}\mathbf{b_2}\right) \mathbf{j} \\ &+ \left(\mathbf{a_1} \circ \mathbf{d_2} + \mathbf{b_1} \circ \mathbf{c_2} - \mathbf{c_1} \circ \mathbf{b_2} + \mathbf{d_1} \circ \mathbf{a_2}\right) \mathbf{k}.\end{aligned} \tag{29}$$

The Hadmard product (Eq. 30) and circular correlation (Eq. 31) are utilized in semantic matching based methods (Sec. III-B2). Hadmard product, denoted as $\circ$ or $\odot : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, is also known as element-wise product or Schur product.

$$(\mathbf{h} \circ \mathbf{t})_i = (\mathbf{h} \odot \mathbf{t})_i = (\mathbf{h})_i (\mathbf{t})_i \tag{30}$$

Circular correlation $\star : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is an efficient computation calculated as:

$$[\boldsymbol{a} \star \boldsymbol{b}]_k = \sum_{i=0}^{d-1} a_i b_{(k+i) \bmod d}. \tag{31}$$

Appendix B
A Summary of KRL Models

We conduct a comprehensive summary of KRL models in Table V. The representation space has an impact on the expressiveness of KRL methods to some extent. By expanding point-wise Euclidean space [15], [17], [20], manifold space [27], complex space [22]–[24] and Gaussian distribution [25], [26] are introduced. ManifoldE [27] relaxes the real-valued point-wise space into manifold space with more expressive representation from the geometric perspective. When $\mathcal{M}(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$ and D_r is set to be zero, the

TABLE V: A comprehensive summary of knowledge representation learning models

Category	Model	Ent. embed.	Rel. embed.	Scoring Function $f_r(h,t)$
Polar coordinate	HAKE [18]	$\mathbf{h}_m, \mathbf{t}_m \in \mathbb{R}^k$ $\mathbf{h}_p, \mathbf{t}_p \in [0, 2\pi)^k$	$\mathbf{r}_m \in \mathbb{R}_+^k$ $\mathbf{r}_p, \in [0, 2\pi)^k$	$-\lVert \mathbf{h}_m \circ \mathbf{r}_m - \mathbf{t}_m \rVert_2 -$ $\lambda \lVert \sin\left(\left(\mathbf{h}_p + \mathbf{r}_p - \mathbf{t}_p\right)/2\right) \rVert_1$
Complex vector	ComplEx [22]	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$\mathrm{Re}\left(< \mathbf{r}, \mathbf{h}, \bar{\mathbf{t}} >\right) = \mathrm{Re}\left(\sum_{k=1}^{K} \mathbf{r}_k \mathbf{h}_k \bar{\mathbf{t}}_k\right)$
	RotatE [23]	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$\lVert \mathbf{h} \circ \mathbf{r} - \mathbf{t} \rVert$
	QuatE [24]	$\mathbf{h}, \mathbf{t} \in \mathbb{H}^d$	$\mathbf{r} \in \mathbb{H}^d$	$\mathbf{h} \otimes \frac{\mathbf{r}}{\lvert \mathbf{r} \rvert} \cdot \mathbf{t}$
Manifold & Group	ManifoldE [27]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\left\lVert \mathcal{M}(h, r, t) - D_r^2 \right\rVert^2$
	TorusE [30]	$[\mathbf{h}], [\mathbf{t}] \in \mathbb{T}^n$	$[\mathbf{r}] \in \mathbb{T}^n$	$\min_{(x,y) \in ([h]+[r]) \times [t]} \lVert x - y \rVert_i$
	DihEdral [31]	$\mathbf{h}^{(l)}, \mathbf{t}^{(l)} \in \mathbb{R}^2$	$\mathbf{R}^{(l)} \in \mathbb{D}_K$	$\sum_{l=1}^{L} \mathbf{h}^{(l)\top} \mathbf{R}^{(l)} \mathbf{t}^{(l)}$
	MuRP [28]	$\mathbf{h}, \mathbf{t} \in \mathbb{B}_c^d, b_h, b_t \in \mathbb{R}$	$\mathbf{r} \in \mathbb{B}_c^d$	$-d_{\mathbb{B}}\left(\exp_{\mathbf{0}}^c\left(\mathbf{R} \log_{\mathbf{0}}^c(\mathbf{h})\right), \mathbf{t} \oplus_c \mathbf{r}\right)^2 + b_h + b_t$
	AttH [29]	$\mathbf{h}, \mathbf{t} \in \mathbb{B}_c^d, b_h, b_t \in \mathbb{R}$	$\mathbf{r} \in \mathbb{B}_c^d$	$-d_{\mathbb{B}}^{c_r}\left(\mathrm{Att}\left(\mathbf{q}_{\mathrm{Rot}}^H, \mathbf{q}_{\mathrm{Ref}}^H; \mathbf{a}_r\right) \oplus^{c_r} \mathbf{r}_r^H, \mathbf{e}_t^H\right)^2 + b_h + b_t$
Gaussian	KG2E [25]	$\mathbf{h} \sim \mathcal{N}\left(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h\right)$ $\mathbf{t} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \Sigma_t\right)$	$\mathbf{r} \sim \mathcal{N}\left(\boldsymbol{\mu}_r, \Sigma_r\right)$	$\int_{x \in \mathcal{R}^{k_e}} \mathcal{N}\left(x; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r\right) \log \frac{\mathcal{N}(x; \boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e)}{\mathcal{N}(x; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)} dx$
		$\boldsymbol{\mu}_h, \boldsymbol{\mu}_t \in \mathbb{R}^d$ $\Sigma_h, \Sigma_t \in \mathbb{R}^{d \times d}$	$\boldsymbol{\mu}_r \in \mathbb{R}^d, \Sigma_r \in \mathbb{R}^{d \times d}$	$\log \int_{x \in \mathcal{R}^{k_e}} \mathcal{N}\left(x; \boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e\right) \mathcal{N}\left(x; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r\right) dx$
	TransG [26]	$\mathbf{h} \sim \mathcal{N}\left(\boldsymbol{\mu}_h, \boldsymbol{\sigma}_h^2 \mathbf{I}\right)$ $\mathbf{t} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \Sigma_t\right)$ $\boldsymbol{\mu}_h, \boldsymbol{\mu}_t \in \mathbb{R}^d$	$\boldsymbol{\mu}_r^i \sim \mathcal{N}\left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_h, \left(\sigma_h^2 + \sigma_t^2\right) \mathbf{I}\right)$ $\mathbf{r} = \sum_i \pi_r^i \boldsymbol{\mu}_r^i \in \mathbb{R}^d$	$\sum_i \pi_r^i \exp\left(-\frac{\left\lVert \boldsymbol{\mu}_h + \boldsymbol{\mu}_r^i - \boldsymbol{\mu}_t \right\rVert_2^2}{\sigma_h^2 + \sigma_t^2}\right)$
Translational Distance	TransE [15]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\lVert \mathbf{h} + \mathbf{r} - \mathbf{t} \rVert_{1/2}$
	TransR [16]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^k, \mathbf{M}_r \in \mathbb{R}^{k \times d}$	$-\lVert \mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t} \rVert_2^2$
	TransH [19]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{w}_r \in \mathbb{R}^d$	$-\left\lVert \left(\mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r\right) + \mathbf{r} - \left(\mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r\right) \right\rVert_2^2$
	TransA [34]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d, \mathbf{M}_r \in \mathbb{R}^{d \times d}$	$(\lvert \mathbf{h} + \mathbf{r} - \mathbf{t} \rvert)^\top \mathbf{W}_{\mathbf{r}} (\lvert \mathbf{h} + \mathbf{r} - \mathbf{t} \rvert)$
	TransF [35]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$(\mathbf{h} + \mathbf{r})^\top \mathbf{t} + (\mathbf{t} - \mathbf{r})^\top \mathbf{h}$
	ITransF [36]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\left\lVert \boldsymbol{\alpha}_r^H \cdot \mathbf{D} \cdot \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_r^T \cdot \mathbf{D} \cdot \mathbf{t} \right\rVert_\ell$
	TransAt [37]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$P_r\left(\sigma\left(\mathbf{r}_h\right) \mathbf{h}\right) + \mathbf{r} - P_r\left(\sigma\left(\mathbf{r}_t\right) \mathbf{t}\right)$
	TransD [33]	$\mathbf{h}, \mathbf{t}, \mathbf{w}_h \mathbf{w}_t \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{w}_r \in \mathbb{R}^k$	$-\left\lVert \left(\mathbf{w}_r \mathbf{w}_h^\top + \mathbf{I}\right) \mathbf{h} + \mathbf{r} - \left(\mathbf{w}_r \mathbf{w}_t^\top + \mathbf{I}\right) \mathbf{t} \right\rVert_2^2$
	TransM [192]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\theta_r \lVert \mathbf{h} + \mathbf{r} - \mathbf{t} \rVert_{1/2}$
	TranSparse [193]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^k, \mathbf{M}_r\left(\theta_r\right) \in \mathbb{R}^{k \times d}$ $\mathbf{M}_r^1\left(\theta_r^1\right), \mathbf{M}_r^2\left(\theta_r^2\right) \in \mathbb{R}^{k \times d}$	$-\lVert \mathbf{M}_r\left(\theta_r\right) \mathbf{h} + \mathbf{r} - \mathbf{M}_r\left(\theta_r\right) \mathbf{t} \rVert_{1/2}^2$ $-\left\lVert \mathbf{M}_r^1\left(\theta_r^1\right) \mathbf{h} + \mathbf{r} - \mathbf{M}_r^2\left(\theta_r^2\right) \mathbf{t} \right\rVert_{1/2}^2$
Semantic Matching	TATEC [194]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d, \mathbf{M}_r \in \mathbb{R}^{d \times d}$	$\mathbf{h}^\top \mathbf{M}_r \mathbf{t} + \mathbf{h}^\top \mathbf{r} + \mathbf{t}^\top \mathbf{r} + \mathbf{h}^\top \mathbf{D} \mathbf{t}$
	ANALOGY [21]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^{d \times d}$	$\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$
	CrossE [41]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\sigma\left(\tanh\left(\mathbf{c}_r \circ \mathbf{h} + \mathbf{c}_r \circ \mathbf{h} \circ \mathbf{r} + \mathbf{b}\right) \mathbf{t}^\top\right)$
	SME [39]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$g_{\text{left}}(\mathbf{h}, \mathbf{r})^\top g_{\text{right}}(\mathbf{r}, \mathbf{t})$
	DistMult [32]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\mathbf{h}^\top \operatorname{diag}(\mathbf{M}_r) \mathbf{t}$
	HolE [20]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\mathbf{r}^\top (h \star t)$
	HolEx [40]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\sum_{j=0}^{l} p\left(\mathbf{h}, \boldsymbol{r}; \boldsymbol{c}_j\right) \cdot \boldsymbol{t}$
	SE [8]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r^1, \mathbf{M}_r^2 \in \mathbb{R}^{d \times d}$	$-\left\lVert \mathbf{M}_r^1 \mathbf{h} - \mathbf{M}_r^2 \mathbf{t} \right\rVert_1$
	SimplE [47]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{r}' \in \mathbb{R}^d$	$\frac{1}{2}\left(\mathbf{h} \circ \mathbf{r} \mathbf{t} + \mathbf{t} \circ \mathbf{r}' \mathbf{t}\right)$
	RESCAL [48]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^{d \times d}$	$\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$
	LFM [50]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{u}_r, \mathbf{v}_r \in \mathbb{R}^p$	$\mathbf{h}^\top \sum_{i=1}^{d} \boldsymbol{\alpha}_i^r \mathbf{u}_i \mathbf{v}_i^\top \mathbf{t}$
	TuckER [51]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}_e^d$	$\mathbf{r} \in \mathbb{R}_r^d$	$\mathcal{W} \times_1 \mathbf{h} \times_2 \mathbf{r} \times_3 \mathbf{t}$
	LowFER [52]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\left(\mathbf{S}^k \operatorname{diag}\left(\mathbf{U}^T \mathbf{h}\right) \mathbf{V}^T \mathbf{r}\right)^T \mathbf{t}$
Neural Networks	MLP [3]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\sigma(\mathbf{w}^\top \sigma(\mathbf{W}[\mathbf{h}, \mathbf{r}, \mathbf{t}]))$
	NAM [53]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\sigma\left(\mathbf{z}^{(L)} \cdot \mathbf{t} + \mathbf{B}^{(L+1)} \mathbf{r}\right)$
	ConvE [54]	$\mathbf{M}_h \in \mathbb{R}^{d_w \times d_h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^{d_w \times d_h}$	$\sigma\left(\operatorname{vec}\left(\sigma\left(\left[\mathbf{M}_h; \mathbf{M}_r\right] * \boldsymbol{\omega}\right)\right) \mathbf{W}\right) \mathbf{t}$
	ConvKB [42]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\operatorname{concat}\left(\sigma\left(\left[\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}\right] * \boldsymbol{\omega}\right)\right) \cdot \mathbf{w}$
	HypER [55]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{w}_r \in \mathbb{R}^{d_r}$	$\sigma\left(\operatorname{vec}\left(\mathbf{h} * \operatorname{vec}^{-1}\left(\mathbf{w}_r \mathbf{H}\right)\right) \mathbf{W}\right) \mathbf{t}$
	SACN [43]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$g\left(\operatorname{vec}\left(\mathbf{M}\left(\mathbf{h}, \mathbf{r}\right)\right) W\right) \mathbf{t}$
	NTN [17]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{b}_r \in \mathbb{R}^k, \widehat{\mathbf{M}} \in \mathbb{R}^{d \times d \times k}$ $\mathbf{M}_{r,1}, \mathbf{M}_{r,2} \in \mathbb{R}^{k \times d}$	$\mathbf{r}^\top \sigma\left(\mathbf{h}^T \widehat{\mathbf{M}} \mathbf{t} + \mathbf{M}_{r,1} \mathbf{h} + \mathbf{M}_{r,2} \mathbf{t} + \mathbf{b}_r\right)$

manifold collapses into a point. With the introduction of the rotational Hadmard product, RotatE [23] can also capture inversion and composition patterns as well as symmetry and antisymmetry. QuatE [24] uses Hamilton product to capture latent inter-dependency within four-dimensional space of entities and relations, and gains more expressive rotational capability than RotatE. Group theory remains less explored to capture rich information of relations. The very recent DihEdral [31] firstly introduces the finite non-Abelian group to preserve the relational properties of symmetry/skew-symmetry, inversion, and composition effectively with the rotation and reflection properties in the dihedral group. Ebisu and Ichise [30] summarized that the embedding space should follow three conditions, i.e., differentiability, calculation possibility, and definability of a scoring function.

Distance-based and semantic matching scoring functions consist of the foundation stones of plausibility measure in KRL. Translational distance-based methods, especially the groundbreaking TransE [15], borrowed the idea of distributed word representation learning and inspired many following approaches, such as TransH [19] and TransR [16] which specify complex relations (1-to-N, N-to-1, and N-to-N) and the recent TransMS [38] which models multi-directional semantics. As for the semantic matching side, many methods utilizes mathematical operations or compositional operators including linear matching in SME [39], bilinear mapping in DistMult [32], tensor product in NTN [17], circular correlation in HolE [20] and ANALOGY [21], Hadamard product in CrossE [41], and quaternion inner product in QuatE [24].

Recent encoding models for knowledge representation have developed rapidly and generally fall into two families of bilinear and neural networks. Linear and bilinear models use product-based functions over entities and relations, while factorization models regard knowledge graphs as three-way tensors. With the multiplicative operations, RESCAL [48], ComplEx [22], and SimplE [47] also belong to the bilinear models. DistMult [32] can only model symmetric relations, while its extension of ComplEx [22] managed to preserve antisymmetric relations, but involves redundant computations [47]. ComplEx [22], SimplE [47], and TuckER [51] can guarantee full expressiveness under specific embedding dimensionality bounds. Neural network-based encoding models start from distributed representation of entities and relations, and some utilizes complex neural structures such as tensor networks [17], graph convolution networks [43], [59], [61], recurrent networks [44] and transformers [45], [58] to learn richer representation. These deep models have achieved very competitive results, but they are not transparent, and lack of interpretability. As deep learning techniques are growing prosperity and gaining extensive superiority in many tasks, the recent trend is still likely to focus on more powerful neural architectures or large-scale pre-training, while deep interpretable models remains a challenge.

Appendix C
KRL Model Training

Open world assumption (OWA) and closed world assumption (CWA) [195] are considered when training knowledge representation learning models. During training, a negative sample set $\mathcal{F}'$ is randomly generated by corrupting a golden triple set $\mathcal{F}$ under the OWA. Mini-batch optimization and Stochastic Gradient Descent (SGD) are carried out to minimize a certain loss function. Under the OWA, negative samples are generated with specific sampling strategies designed to reduce the number of false negatives.

A. Open and Closed World Assumption

The CWA assumes that unobserved facts are false. By contrast, the OWA has a relaxed assumption that unobserved ones can be either missing or false. Generally, OWA has an advantage over CWA because of the incompleteness nature of knowledge graphs. RESCAL [48] is a typical model trained under the CWA, while more models are formulated under the OWA.

B. Loss Function

Several families of loss function are introduced for KRL model optimization. First, a margin-based loss is optimized to learn representations that positive samples have higher scores than negative ones. Some literature also called it as pairwise ranking loss. As shown in Eq. 32 , the rank-based hinge loss maximizes the discriminative margin between a golden triple (h, r, t) and an invalid triple (h', r, t').

$$\min_{\Theta} \sum_{(h,r,t)\in\mathcal{F}} \sum_{(h',r,t')\in\mathcal{F}'} \max\left(0, f_r(h,t) + \gamma - f_r\left(h', t'\right)\right) \quad (32)$$

here γ is a margin. The invalid triple (h', r, t') is constructed by randomly changing a head or tail entity or both entities in the knowledge graph. Most translation-based embedding methods use margin-based loss [196]. The second kind of loss function is logistic-based loss in Eq. 33, which is to minimize negative log-likelihood of logistic models.

$$\min_{\Theta} \sum_{(h,r,t)\in\mathcal{F}\cup\mathcal{F}'} \log\left(1 + \exp\left(-y_{hrt} \cdot f_r(h,t)\right)\right) \quad (33)$$

here y_{hrt} is the label of triple instance. Some methods also use other kinds of loss functions. For example, ConvE and TuckER use binary cross-entropy or the so-called Bernoulli negative log-likelihood loss function defined as:

$$-\frac{1}{N_e}\sum_{i}^{N_e}\left(y_i \cdot \log\left(p_i\right) + \left(1 - y_i\right) \cdot \log\left(1 - p_i\right)\right), \quad (34)$$

where p is the prediction and y is the ground label. And RotatE uses the form of loss function in Eq. 35.

$$-\log\sigma\left(\gamma - f_r(h,t)\right) - \sum_{i=1}^{n}\frac{1}{k}\log\sigma\left(f_r\left(h_i', t_i'\right) - \gamma\right) \quad (35)$$

For all those kinds of loss functions, specific regularization like L2 on parameters or constraints can also be applied, as well as combined with the joint learning paradigm.

C. Negative Sampling

Several heuristics of sampling distribution are proposed to corrupt the head or tail entities. The widest applied one is uniform sampling [15], [16], [39] that uniformly replaces entities. But it leads to the sampling of false-negative labels. More effective negative sampling strategies are required to learn semantic representation and improve predictive performance.

Considering the mapping property of relations, Bernoulli sampling [19] introduces a heuristic of a sampling distribution as $\frac{tph}{tph+hpt}$, where tph and hpt denote the average number of tail entities per head entity and the average number of head entities per tail entity respectively. Domain sampling [36] chooses corrupted samples from entities in the same domain or from the whole entity set with a relation-dependent probability p_r or $1 - p_r$ respectively, with the head and tail domain of relation r denoted as $\mathrm{M}_r^H = \{h \mid \exists\, t(h,r,t) \in P\}$ and $\mathrm{M}_r^T = \{t \mid \exists\, h(h,r,t) \in P\}$, and induced relational set denoted as $\mathrm{N}_r = \{(h,r,t) \in P\}$.

Recently, two adversarial sampling methods are further proposed. KBGAN [196] introduces adversarial learning for negative sampling, where the generator uses probability-based log-loss embedding models. The probability of generating negative samples $p\left(h_j', r, t_j' \mid \{(h_i, r_i, t_i)\}\right)$ is defined as

$$\frac{\exp f_G\left(h_i', r, t_i'\right)}{\sum_{j=1} \exp f_G\left(h_j', r, t_j'\right)}, \tag{36}$$

where $f_G(h, r, t)$ is the scoring function of generator. Similarly, Sun et al. [23] proposed self-adversarial negative sampling based on self scoring function by sampling negative triples from the distribution in Eq. 37, where α is the temperature of sampling.

$$p\left(h_j', r, t_j' \mid \{(h_i, r_i, t_i)\}\right) = \frac{\exp \alpha f\left(h_j', r, t_j'\right)}{\sum_i \exp \alpha f\left(h_i', r, t_i'\right)} \tag{37}$$

Negative sampling strategies are summarized in Table VI. Trouillon et al. [22] studied the number of negative samples generated per positive training sample and found a trade-off between accuracy and training time.

TABLE VI: A summary of negative sampling

Sampling	Mechanism	Sampling probability
Uniform [39]	uniform distribution	$\frac{1}{n}$
Bernoulli [19]	mapping property	$\frac{tph}{tph+hpt}$
Domain [36]	relation-depend domain	$\min\left(\frac{\lambda\left\|\mathrm{M}_r^T\right\|\left\|\mathrm{M}_r^H\right\|}{\|N_r\|}, 0.5\right)$
Adversarial [196]	generator embedding	$\frac{\exp f_G\left(h_i', r, t_i'\right)}{\sum_{j=1} \exp f_G\left(h_j', r, t_j'\right)}$
Self-adversarial [23]	current embedding	$\frac{\exp \alpha f\left(h_j', r, t_j'\right)}{\sum_i \exp \alpha f\left(h_i', r, t_i'\right)}$

Appendix D
More Knowledge-aware Applications

There are also many other applications that utilize knowledge-driven methods. *1) Question generation* focuses on generating natural language questions. Seyler et al. [197] studied quiz-style knowledge question generation by generating a structured triple-pattern query over the knowledge graph while estimating how difficult the questions are. But for verbalizing the question, the authors used a template-based method, which may have a limitation on generating more natural expression. *2) Academic search engine* helps research to find relevant academic papers. Xiong et al. [198] proposed explicit semantic ranking with knowledge graph embedding to help academic search better understand the meaning of query concepts. *3) Medical applications* involve with domain-specific knowledge graph of medical concepts. Li et al. [199] formulated medical image report generation by three steps of encoding, retrieval, and paraphrasing, where the medical image is encoded by the abnormality graph. *4) Mental healthcare* with knowledge graph facilitates a good understanding of mental conditions and risk factors of mental disorders and is applied to effective prevention of mental health leaded suicide. Gaurs et al. [200] developed a rule-based classifier for knowledge-aware suicide risk assessment with a suicide risk severity lexicon incorporating medical knowledge bases and suicide ontology. *5) Zero-shot image classification* gets benefits from knowledge graph propagation with semantic descriptions of classes. Wang et al. [201] proposed a multi-layer GCN to learn zero-shot classifiers using semantic embeddings of categories and categorical relationships. APNet [202] propagates attribute representations with category graph. *6) Text generation* synthesizes and composes coherent multi-sentence texts. Koncel-Kedziorski et al. [203] studied text generation for information extraction systems and proposed a graph transforming encoder for graph-to-text generation from the knowledge graph. *7) Sentiment analysis* integrated with sentiment-related concepts can better understand people's opinions and sentiments. SenticNet [204] learns conceptual primitives for sentiment analysis, which can also be used as a commonsense knowledge source. To enable sentiment-related information filtering, Sentic LSTM [205] injects knowledge concepts to the vanilla LSTM and designs a knowledge output gate for concept-level output as a complement to the token level.

A. Dialogue Systems

QA can also be viewed as a single-turn dialogue system by generating the correct answer as a response, while dialogue systems consider conversational sequences and aim to generate fluent responses to enable multi-round conversations via semantic augmentation and knowledge graph walk. Liu et al. [206] encoded knowledge to augment semantic representation and generated knowledge aware response by knowledge graph retrieval and graph attention mechanism under an encoder-decoder framework. DialKG Walker [207] traverses a symbolic knowledge graph to learn contextual transition in dialogue and predicts entity responses with attentive graph path decoder.

Semantic parsing via formal logical representation is another direction for dialog systems. By predefining a set of base actions, Dialog-to-Action [208] is an encoder-decoder approach that maps executable logical forms from the utterance in conversation, to generate action sequence under the control of a grammar-guided decoder.

Appendix E
Datasets and Libraries

In this section, we introduce and list useful resources of knowledge graph datasets and open-source libraries.

A. Datasets

Many public datasets have been released. We conduct an introduction and a summary of general, domain-specific, task-specific, and temporal datasets.

1) General Datasets: Datasets with general ontological knowledge include WordNet [209], Cyc [210], DBpedia [212], YAGO [211], Freebase [213], NELL [214] and Wikidata [215]. It is hard to compare them within a table as their ontologies are different. Thus, only an informal comparison is illustrated in Table VII, where their volumes kept going after their release.

WordNet, firstly released in 1995, is a lexical database that contains about 117,000 synsets. DBpedia is a community-driven dataset extracted from Wikipedia. It contains 103 million triples and can be enlarged when interlinked with other open datasets. To solve the problems of low coverage and low quality of single-source ontological knowledge, YAGO utilized the

TABLE VII: Statistics of datasets with general knowledge when originally released

Dataset	# entities	# facts	Website
WordNet [209]	117,597	207,016	https://wordnet.princeton.edu
OpenCyc [210]	47,000	306,000	https://www.cyc.com/opencyc/
Cyc [210]	∼250,000	∼2,200,000	https://www.cyc.com
YAGO [211]	1,056,638	∼5,000,000	http://www.mpii.mpg.de/∼suchanek/yago
DBpedia [212]	∼1,950,000	∼103,000,000	https://wiki.dbpedia.org/develop/datasets
Freebase [213]	-	∼125,000,000	https://developers.google.com/freebase/
NELL [214]	-	242,453	http://rtw.ml.cmu.edu/rtw/
Wikidata [215]	14,449,300	30,263,656	https://www.wikidata.org/wiki
Probase IsA	12,501,527	85,101,174	https://concept.research.microsoft.com/Home/Download
Google KG	> 500 million	> 3.5 billion	https://developers.google.com/knowledge-graph

TABLE VIII: A summary of datasets for tasks on knowledge graph itself

Dataset	# Rel.	#Ent.	# Train	# Valid.	# Test
WN18 [15]	18	40,943	141,442	5,000	5,000
FB15K [15]	1,345	14,951	483,142	50,000	59,071
WN11 [17]	11	38,696	112,581	2,609	10,544
FB13 [17]	13	75,043	316,232	5,908	23,733
WN18RR [54]	11	40,943	86,835	3,034	3,134
FB15k-237 [216]	237	14,541	272,115	17,535	20,466
FB5M [19]	1,192	5,385,322	19,193,556	50,000	59,071
FB40K [16]	1,336	39,528	370,648	67,946	96,678

concept information in the category page of Wikipedia and the hierarchy information of concepts in WordNet to build a multi-source dataset with high coverage and quality. Moreover, it is extendable by other knowledge sources. It is available online with more than 10 million entities and 120 million facts currently. Freebase, a scalable knowledge base, came up for the storage of the world's knowledge in 2008. Its current number of triples is 1.9 billion. NELL is built from the Web via an intelligent agent called Never-Ending Language Learner. It has 2,810,379 beliefs with high confidence by far. Wikidata is a free structured knowledge base, which is created and maintained by human editors to facilitate the management of Wikipedia data. It is multi-lingual with 358 different languages.

The aforementioned datasets are openly published and maintained by communities or research institutions. There are also some commercial datasets. The Cyc knowledge base from Cycorp contains about 1.5 million general concepts and more than 20 million general rules, with an accessible version called OpenCyc deprecated sine 2017. Google knowledge graph hosts more than 500 million entities and 3.5 billion facts and relations. Microsoft builds a probabilistic taxonomy called Probase [217] with 2.7 million concepts.

2) Domain-Specific Datasets: Some knowledge bases on specific domains are designed and collected to evaluate domain-specific tasks. Some notable domains include life science, health care, and scientific research, covering complex domains and relations such as compounds, diseases, and tissues. Examples of domain-specific knowledge graphs are ResearchSpace[6], a cultural heritage knowledge graph; UMLS [218], a unified medical language system; GeneOntology[7], a gene ontology resource; SNOMED CT[8], a commercial clinical terminology; and a medical knowledge graph from Yidu Research[9].

3) Task-Specific Datasets: A popular way of generating task-specific datasets is to sample subsets from large general datasets. Statistics of several datasets for tasks on the knowledge graph itself are listed in Table VIII. Notice that WN18 and FB15k suffer from test set leakage [54]. For KRL with auxiliary information and other downstream knowledge-aware applications, texts and images are also collected, for example, WN18-IMG [70] with sampled images and textual relation extraction dataset including SemEval 2010 dataset, NYT [219] and Google-RE[10]. IsaCore [220], an analogical closure of Probase for opinion mining and sentiment analysis, is built by common knowledge base blending and multi-dimensional scaling. Recently, the FewRel dataset [221] was built to evaluate the emerging few-shot relation classification task. There are also more datasets for specific tasks such as cross-lingual DBP15K [115] and DWY100K [114] for entity alignment, multi-view knowledge graphs of YAGO26K-906 and DB111K-174 [222] with instances and ontologies.

Numerous downstream knowledge-aware applications also come up with many datasets, for example, WikiFacts [223] for language modeling; SimpleQuestions [174] and LC-QuAD [224] for question answering; and Freebase Semantic Scholar [198] for academic search.

B. Open-Source Libraries

TABLE IX: A summary of open-source libraries

Task	Library	Language	URL
General	Grakn	Python	github.com/graknlabs/kglib
General	AmpliGraph	TensorFlow	github.com/Accenture/AmpliGraph
General	GraphVite	Python	graphvite.io
Database	Akutan	Go	github.com/eBay/akutan
KRL	OpenKE	PyTorch	github.com/thunlp/OpenKE
KRL	Fast-TransX	C++	github.com/thunlp/Fast-TransX
KRL	scikit-kge	Python	github.com/mnick/scikit-kge
KRL	LibKGE	PyTorch	github.com/uma-pi1/kge
KRL	PyKEEN	Python	github.com/SmartDataAnalytics/PyKEEN
RE	OpenNRE	PyTorch	github.com/thunlp/OpenNRE

Recent research has boosted the open-source campaign, with several libraries listed in Table IX. They are AmpliGraph [225]

[6] https://www.researchspace.org/index.html

[7] http://geneontology.org

[8] http://www.snomed.org/snomed-ct/five-step-briefing

[9] https://www.yiducloud.com.cn/en/academy.html

[10] https://code.google.com/archive/p/relation-extraction-corpus/

for knowledge representation learning, Grakn for integration knowledge graph with machine learning techniques, and Akutan for knowledge graph store and query. The research community has also released codes to facilitate further research. Notably, there are three useful toolkits, namely scikit-kge and OpenKE [226] for knowledge graph embedding, and OpenNRE [227] for relation extraction. We provide an online collection of knowledge graph publications, together with links to some open-source implementations of them, hosted at https://shaoxiongji.github.io/knowledge-graphs/.

References

[1] A. Newell, J. C. Shaw, and H. A. Simon, "Report on a general problem solving program," in *IFIP congress*, vol. 256, 1959, p. 64.

[2] E. Shortliffe, *Computer-based medical consultations: MYCIN*. Elsevier, 2012, vol. 2.

[3] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *SIGKDD*. ACM, 2014, pp. 601–610.

[4] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2016.

[5] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE TKDE*, vol. 29, no. 12, pp. 2724–2743, 2017.

[6] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres *et al.*, "Knowledge graphs," *arXiv preprint arXiv:2003.02320*, 2020.

[7] F. N. Stokman and P. H. de Vries, "Structuring knowledge in a graph," in *Human-Computer Interaction*, 1988, pp. 186–206.

[8] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *AAAI*, 2011, pp. 301–306.

[9] Y. Lin, X. Han, R. Xie, Z. Liu, and M. Sun, "Knowledge representation learning: A quantitative review," *arXiv preprint arXiv:1812.10901*, 2018.

[10] R. H. Richens, "Preprogramming for mechanical translation." *Mechanical Translation*, vol. 3, no. 1, pp. 20–25, 1956.

[11] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.

[12] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs," *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, pp. 1–4, 2016.

[13] T. Wu, G. Qi, C. Li, and M. Wang, "A survey of techniques for constructing chinese knowledge graphs and their applications," *Sustainability*, vol. 10, no. 9, p. 3245, 2018.

[14] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, p. 112948, 2020.

[15] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS*, 2013, pp. 2787–2795.

[16] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *AAAI*, 2015, pp. 2181–2187.

[17] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *NIPS*, 2013, pp. 926–934.

[18] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction." in *AAAI*, 2020, pp. 3065–3072.

[19] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *AAAI*, 2014, pp. 1112–1119.

[20] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *AAAI*, 2016, pp. 1955–1961.

[21] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," in *ICML*, 2017, pp. 2168–2178.

[22] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *ICML*, 2016, pp. 2071–2080.

[23] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," in *ICLR*, 2019, pp. 1–18.

[24] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embedding," in *NeurIPS*, 2019, pp. 2731–2741.

[25] S. He, K. Liu, G. Ji, and J. Zhao, "Learning to represent knowledge graphs with gaussian embedding," in *CIKM*, 2015, pp. 623–632.

[26] H. Xiao, M. Huang, and X. Zhu, "TransG: A generative model for knowledge graph embedding," in *ACL*, vol. 1, 2016, pp. 2316–2325.

[27] ——, "From one point to a manifold: Orbit models for knowledge graph embedding," in *IJCAI*, 2016, pp. 1315–1321.

[28] I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational poincaré graph embeddings," in *NeurIPS*, 2019, pp. 4463–4473.

[29] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in *ACL*, 2020.

[30] T. Ebisu and R. Ichise, "TorusE: Knowledge graph embedding on a lie group," in *AAAI*, 2018, pp. 1819–1826.

[31] C. Xu and R. Li, "Relation embedding with dihedral group in knowledge graph," in *ACL*, 2019, pp. 263–272.

[32] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *ICLR*, 2015, pp. 1–13.

[33] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *ACL-IJCNLP*, vol. 1, 2015, pp. 687–696.

[34] H. Xiao, M. Huang, Y. Hao, and X. Zhu, "TransA: An adaptive approach for knowledge graph embedding," in *AAAI*, 2015, pp. 1–7.

[35] J. Feng, M. Huang, M. Wang, M. Zhou, Y. Hao, and X. Zhu, "Knowledge graph embedding by flexible translation," in *KR*, 2016, pp. 557–560.

[36] Q. Xie, X. Ma, Z. Dai, and E. Hovy, "An interpretable knowledge transfer model for knowledge base completion," in *ACL*, 2017, pp. 950–962.

[37] W. Qian, C. Fu, Y. Zhu, D. Cai, and X. He, "Translating embeddings for knowledge graph completion with relation attention mechanism." in *IJCAI*, 2018, pp. 4286–4292.

[38] S. Yang, J. Tian, H. Zhang, J. Yan, H. He, and Y. Jin, "TransMS: knowledge graph embedding for complex relations by multidirectional semantics," in *IJCAI*, 2019, pp. 1935–1942.

[39] A. Bordes, X. Glorot, 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.

[40] Y. Xue, Y. Yuan, Z. Xu, and A. Sabharwal, "Expanding holographic embeddings for knowledge completion," in *NeurIPS*, 2018, pp. 4491–4501.

[41] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in *WSDM*, 2019, pp. 96–104.

[42] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *NAACL*, 2018, pp. 327–333.

[43] 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 *AAAI*, vol. 33, 2019, pp. 3060–3067.

[44] L. Guo, Z. Sun, and W. Hu, "Learning to exploit long-term relational dependencies in knowledge graphs," in *ICML*, 2019, pp. 2505–2514.

[45] Q. Wang, P. Huang, H. Wang, S. Dai, W. Jiang, J. Liu, Y. Lyu, Y. Zhu, and H. Wu, "CoKE: Contextualized knowledge graph embedding," *arXiv preprint arXiv:1911.02168*, 2019.

[46] Y. Wang, R. Gemulla, and H. Li, "On multi-relational link prediction with bilinear models," in *AAAI*, 2018, pp. 4227–4234.

[47] S. M. Kazemi and D. Poole, "SimplE embedding for link prediction in knowledge graphs," in *NeurIPS*, 2018, pp. 4284–4295.

[48] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *ICML*, vol. 11, 2011, pp. 809–816.

[49] ——, "Factorizing YAGO: scalable machine learning for linked data," in *WWW*, 2012, pp. 271–280.

[50] R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski, "A latent factor model for highly multi-relational data," in *NIPS*, 2012, pp. 3167–3175.

[51] I. Balažević, C. Allen, and T. M. Hospedales, "TuckER: Tensor factorization for knowledge graph completion," in *EMNLP-IJCNLP*, 2019, pp. 5185–5194.

[52] S. Amin, S. Varanasi, K. A. Dunfield, and G. Neumann, "LowFER: Low-rank bilinear pooling for link prediction," in *ICML*, 2020, pp. 1–11.

[53] Q. Liu, H. Jiang, A. Evdokimov, Z.-H. Ling, X. Zhu, S. Wei, and Y. Hu, "Probabilistic reasoning via deep learning: Neural association models," *arXiv preprint arXiv:1603.07704*, 2016.

[54] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *AAAI*, vol. 32, 2018, pp. 1811–1818.

[55] I. Balažević, C. Allen, and T. M. Hospedales, "Hypernetwork knowledge graph embeddings," in *ICANN*, 2019, pp. 553–565.

[56] M. Gardner, P. Talukdar, J. Krishnamurthy, and T. Mitchell, "Incorporating vector space similarity in random walk inference over knowledge bases," in *EMNLP*, 2014, pp. 397–406.

[57] A. Neelakantan, B. Roth, and A. McCallum, "Compositional vector space models for knowledge base completion," in *ACL-IJCNLP*, vol. 1, 2015, pp. 156–166.

[58] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for knowledge graph completion," *arXiv preprent arXiv:1909.03193*, 2019.

[59] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *ESWC*, 2018, pp. 593–607.

[60] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017, pp. 1–14.

[61] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *ACL*, 2019, pp. 4710–4723.

[62] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *ICLR*, 2020, pp. 1–15.

[63] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph and text jointly embedding," in *EMNLP*, 2014, pp. 1591–1601.

[64] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *AAAI*, 2016, pp. 2659–2665.

[65] H. Xiao, M. Huang, L. Meng, and X. Zhu, "SSP: semantic space projection for knowledge graph embedding with text descriptions," in *AAAI*, 2017, pp. 3104–3110.

[66] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, "Semantically smooth knowledge graph embedding," in *ACL-IJCNLP*, vol. 1, 2015, pp. 84–94.

[67] R. Xie, Z. Liu, and M. Sun, "Representation learning of knowledge graphs with hierarchical types," in *IJCAI*, 2016, pp. 2965–2971.

[68] Y. Lin, Z. Liu, and M. Sun, "Knowledge representation learning with entities, attributes and relations," in *IJCAI*, 2016, pp. 2866–2872.

[69] Z. Zhang, F. Zhuang, M. Qu, F. Lin, and Q. He, "Knowledge graph embedding with hierarchical relation structure," in *EMNLP*, 2018, pp. 3198–3207.

[70] R. Xie, Z. Liu, H. Luan, and M. Sun, "Image-embodied knowledge representation learning," in *IJCAI*, 2017, pp. 3140–3146.

[71] X. Han, Z. Liu, and M. Sun, "Neural knowledge acquisition via mutual attention between knowledge graph and text," in *AAAI*, 2018, pp. 4832–4839.

[72] T. Dong, Z. Wang, J. Li, C. Bauckhage, and A. B. Cremers, "Triple classification using regions and fine-grained entity typing," in *AAAI*, vol. 33, 2019, pp. 77–85.

[73] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *ACL*, vol. 2, 2016, pp. 207–212.

[74] E. Cao, D. Wang, J. Huang, and W. Hu, "Open knowledge enrichment for long-tail entities," in *The Web Conference*, 2020, pp. 384–394.

[75] B. Shi and T. Weninger, "ProjE: Embedding projection for knowledge graph completion," in *AAAI*, 2017, pp. 1236–1242.

[76] S. Guan, X. Jin, Y. Wang, and X. Cheng, "Shared embedding based neural networks for knowledge graph completion," in *CIKM*, 2018, pp. 247–256.

[77] B. Shi and T. Weninger, "Open-world knowledge graph completion," in *AAAI*, 2018, pp. 1957–1964.

[78] C. Zhang, Y. Li, N. Du, W. Fan, and P. S. Yu, "On the generative discovery of structured medical knowledge," in *SIGKDD*, 2018, pp. 2720–2728.

[79] 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, 2010.

[80] R. Das, A. Neelakantan, D. Belanger, and A. McCallum, "Chains of reasoning over entities, relations, and text using recurrent neural networks," in *EACL*, vol. 1, 2017, pp. 132–141.

[81] W. Chen, W. Xiong, X. Yan, and W. Y. Wang, "Variational knowledge graph reasoning," in *NAACL*, 2018, pp. 1823–1832.

[82] W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A reinforcement learning method for knowledge graph reasoning," in *EMNLP*, 2017, pp. 564–573.

[83] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola, and A. McCallum, "Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning," in *ICLR*, 2018, pp. 1–18.

[84] X. V. Lin, R. Socher, and C. Xiong, "Multi-hop knowledge graph reasoning with reward shaping," in *EMNLP*, 2018, pp. 3243–3253.

[85] Y. Shen, J. Chen, P.-S. Huang, Y. Guo, and J. Gao, "M-Walk: Learning to walk over graphs using monte carlo tree search," in *NeurIPS*, 2018, pp. 6786–6797.

[86] C. Fu, T. Chen, M. Qu, W. Jin, and X. Ren, "Collaborative policy learning for open knowledge graph reasoning," in *EMNLP*, 2019, pp. 2672–2681.

[87] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "AMIE: association rule mining under incomplete evidence in ontological knowledge bases," in *WWW*, 2013, pp. 413–422.

[88] P. G. Omran, K. Wang, and Z. Wang, "An embedding-based approach to rule learning in knowledge graphs," *IEEE TKDE*, pp. 1–12, 2019.

[89] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *EMNLP*, 2016, pp. 192–202.

[90] ——, "Knowledge graph embedding with iterative guidance from soft rules," in *AAAI*, 2018, pp. 4816–4823.

[91] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, and H. Chen, "Iteratively learning embeddings and rules for knowledge graph reasoning," in *WWW*, 2019, pp. 2366–2377.

[92] T. Rocktäschel and S. Riedel, "End-to-end differentiable proving," in *NIPS*, 2017, pp. 3788–3800.

[93] F. Yang, Z. Yang, and W. W. Cohen, "Differentiable learning of logical rules for knowledge base reasoning," in *NIPS*, 2017, pp. 2319–2328.

[94] P.-W. Wang, D. Stepanova, C. Domokos, and J. Z. Kolter, "Differentiable learning of numerical rules in knowledge graphs," in *ICLR*, 2020, pp. 1–12.

[95] M. Qu and J. Tang, "Probabilistic logic neural networks for reasoning," in *NeurIPS*, 2019, pp. 7710–7720.

[96] Y. Zhang, X. Chen, Y. Yang, A. Ramamurthy, B. Li, Y. Qi, and L. Song, "Efficient probabilistic logic reasoning with graph neural networks," in *ICLR*, 2020, pp. 1–20.

[97] W. Xiong, M. Yu, S. Chang, X. Guo, and W. Y. Wang, "One-shot relational learning for knowledge graphs," in *EMNLP*, 2018, pp. 1980–1990.

[98] X. Lv, Y. Gu, X. Han, L. Hou, J. Li, and Z. Liu, "Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations," in *EMNLP-IJCNLP*, 2019, pp. 3374–3379.

[99] M. Chen, W. Zhang, W. Zhang, Q. Chen, and H. Chen, "Meta relational learning for few-shot link prediction in knowledge graphs," in *EMNLP-IJCNLP*, 2019, pp. 4217–4226.

[100] C. Zhang, H. Yao, C. Huang, M. Jiang, Z. Li, and N. V. Chawla, "Few-shot knowledge graph completion," in *AAAI*, 2020, pp. 1–8.

[101] P. Qin, X. Wang, W. Chen, C. Zhang, W. Xu, and W. Y. Wang, "Generative adversarial zero-shot relational learning for knowledge graphs," in *AAAI*, 2020, pp. 1–8.

[102] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Transactions of ACL*, vol. 4, pp. 357–370, 2016.

[103] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *NAACL*, 2016, pp. 260–270.

[104] C. Xia, C. Zhang, T. Yang, Y. Li, N. Du, X. Wu, W. Fan, F. Ma, and P. Yu, "Multi-grained named entity recognition," in *ACL*, 2019, pp. 1430–1440.

[105] A. Hu, Z. Dou, J.-Y. Nie, and J.-R. Wen, "Leveraging multi-token entities in document-level named entity recognition." in *AAAI*, 2020, pp. 7961–7968.

[106] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, "A unified MRC framework for named entity recognition," in *ACL*, 2020, pp. 5849–5859.

[107] X. Ren, W. He, M. Qu, C. R. Voss, H. Ji, and J. Han, "Label noise reduction in entity typing by heterogeneous partial-label embedding," in *SIGKDD*, 2016, pp. 1825–1834.

[108] Y. Ma, E. Cambria, and S. Gao, "Label embedding for zero-shot fine-grained named entity typing," in *COLING*, 2016, pp. 171–180.

[109] H. Huang, L. Heck, and H. Ji, "Leveraging deep neural networks and knowledge graphs for entity disambiguation," *arXiv preprint arXiv:1504.07678*, 2015.

[110] W. Fang, J. Zhang, D. Wang, Z. Chen, and M. Li, "Entity disambiguation by knowledge and text jointly embedding," in *SIGNLL*, 2016, pp. 260–269.

[111] O.-E. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," in *EMNLP*, 2017, pp. 2619–2629.

[112] P. Le and I. Titov, "Improving entity linking by modeling latent relations between mentions," in *ACL*, vol. 1, 2018, pp. 1595–1604.

[113] H. Zhu, R. Xie, Z. Liu, and M. Sun, "Iterative entity alignment via joint knowledge embeddings," in *IJCAI*, 2017, pp. 4258–4264.

[114] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding." in *IJCAI*, 2018, pp. 4396–4402.

[115] Z. Sun, W. Hu, and C. Li, "Cross-lingual entity alignment via joint attribute-preserving embedding," in *ISWC*, 2017, pp. 628–644.

[116] M. Chen, Y. Tian, K.-W. Chang, S. Skiena, and C. Zaniolo, "Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment," in *IJCAI*, 2018, pp. 3998–4004.

[117] Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, and Y. Qu, "Multi-view knowledge graph embedding for entity alignment," in *IJCAI*, 2019, pp. 5429–5435.

[118] B. D. Trsedya, J. Qi, and R. Zhang, "Entity alignment between knowledge graphs using attribute embeddings," in *AAAI*, vol. 33, 2019, pp. 297–304.

[119] M. Craven, J. Kumlien *et al.*, "Constructing biological knowledge bases by extracting information from text sources," in *ISMB*, vol. 1999, 1999, pp. 77–86.

[120] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *ACL and IJCNLP of the AFNLP*, 2009, pp. 1003–1011.

[121] J. Qu, D. Ouyang, W. Hua, Y. Ye, and X. Zhou, "Discovering correlations between sparse features in distant supervision for relation extraction," in *WSDM*, 2019, pp. 726–734.

[122] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *COLING*, 2014, pp. 2335–2344.

[123] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *ACL Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 39–48.

[124] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *EMNLP*, 2015, pp. 1753–1762.

[125] X. Jiang, Q. Wang, P. Li, and B. Wang, "Relation extraction with multi-instance multi-label convolutional neural networks," in *COLING*, 2016, pp. 1471–1480.

[126] H. Ye, W. Chao, Z. Luo, and Z. Li, "Jointly extracting relations with class ties via effective deep ranking," in *ACL*, vol. 1, 2017, pp. 1810–1820.

[127] W. Zeng, Y. Lin, Z. Liu, and M. Sun, "Incorporating relation paths in neural relation extraction," in *EMNLP*, 2017, pp. 1768–1777.

[128] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths," in *EMNLP*, 2015, pp. 1785–1794.

[129] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," in *ACL*, vol. 1, 2016, pp. 1105–1116.

[130] R. Cai, X. Zhang, and H. Wang, "Bidirectional recurrent convolutional neural network for relation classification," in *ACL*, vol. 1, 2016, pp. 756–765.

[131] Y. Shen and X. Huang, "Attention-based convolutional neural network for semantic relation extraction," in *COLING*, 2016, pp. 2526–2536.

[132] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *ACL*, vol. 1, 2016, pp. 2124–2133.

[133] G. Ji, K. Liu, S. He, and J. Zhao, "Distant supervision for relation extraction with sentence-level attention and entity descriptions," in *AAAI*, 2017, pp. 3060–3066.

[134] X. Han, P. Yu, Z. Liu, M. Sun, and P. Li, "Hierarchical relation extraction with coarse-to-fine grained attention," in *EMNLP*, 2018, pp. 2236–2245.

[135] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *EMNLP*, 2018, pp. 2205–2215.

[136] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," in *ACL*, 2019, pp. 241–251.

[137] N. Zhang, S. Deng, Z. Sun, G. Wang, X. Chen, W. Zhang, and H. Chen, "Long-tail relation extraction via knowledge graph embeddings and graph convolution networks," in *NAACL*, 2019, pp. 3016–3025.

[138] Y. Wu, D. Bamman, and S. Russell, "Adversarial training for relation extraction," in *EMNLP*, 2017, pp. 1778–1783.

[139] P. Qin, X. Weiran, and W. Y. Wang, "DSGAN: Generative adversarial training for distant supervision relation extraction," in *ACL*, vol. 1, 2018, pp. 496–505.

[140] P. Qin, W. Xu, and W. Y. Wang, "Robust distant supervision relation extraction via deep reinforcement learning," in *ACL*, vol. 1, 2018, pp. 2137–2147.

[141] X. Zeng, S. He, K. Liu, and J. Zhao, "Large scaled relation extraction with reinforcement learning," in *AAAI*, 2018, pp. 5658–5665.

[142] J. Feng, M. Huang, L. Zhao, Y. Yang, and X. Zhu, "Reinforcement learning for relation classification from noisy data," in *AAAI*, 2018, pp. 5779–5786.

[143] R. Takanobu, T. Zhang, J. Liu, and M. Huang, "A hierarchical framework for relation extraction with reinforcement learning," in *AAAI*, vol. 33, 2019, pp. 7072–7079.

[144] Y. Huang and W. Y. Wang, "Deep residual learning for weakly-supervised relation extraction," in *EMNLP*, 2017, pp. 1803–1807.

[145] T. Liu, X. Zhang, W. Zhou, and W. Jia, "Neural relation extraction via inner-sentence noise reduction and transfer learning," in *EMNLP*, 2018, pp. 2195–2204.

[146] K. Lei, D. Chen, Y. Li, N. Du, M. Yang, W. Fan, and Y. Shen, "Cooperative denoising for distantly supervised relation extraction," in *COLING*, 2018, pp. 426–436.

[147] H. Jiang, L. Cui, Z. Xu, D. Yang, J. Chen, C. Li, J. Liu, J. Liang, C. Wang, Y. Xiao, and W. Wang, "Relation extraction using supervision from topic knowledge of relation labels," in *IJCAI*, 2019, pp. 5024–5030.

[148] H. Shahbazi, X. Z. Fern, R. Ghaeini, and P. Tadepalli, "Relation extraction with explanation," in *ACL*, 2020, pp. 6488–6494.

[149] T. Gao, X. Han, Z. Liu, and M. Sun, "Hybrid attention-based prototypical networks for noisy few-shot relation classification," in *AAAI*, vol. 33, 2019, pp. 6407–6414.

[150] M. Qu, T. Gao, L.-P. A. Xhonneux, and J. Tang, "Few-shot relation extraction via bayesian meta-learning on relation graphs," in *ICML*, 2020, pp. 1–10.

[151] J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in *WWW*, 2018, pp. 1771–1776.

[152] Y. Ma, V. Tresp, and E. A. Daxberger, "Embedding models for episodic knowledge graphs," *Journal of Web Semantics*, vol. 59, p. 100490, 2019.

[153] S. S. Dasgupta, S. N. Ray, and P. Talukdar, "Hyte: Hyperplane-based temporally aware knowledge graph embedding," in *EMNLP*, 2018, pp. 2001–2011.

[154] A. García-Durán, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in *EMNLP*, 2018, pp. 4816–4821.

[155] Y. Liu, W. Hua, K. Xin, and X. Zhou, "Context-aware temporal knowledge graph embedding," in *WISE*, 2019, pp. 583–598.

[156] T. Lacroix, G. Obozinski, and N. Usunier, "Tensor decompositions for temporal knowledge base completion," in *ICLR*, 2020, pp. 1–12.

[157] D. T. Wijaya, N. Nakashole, and T. M. Mitchell, "CTPs: Contextual temporal profiles for time scoping facts using state change detection," in *EMNLP*, 2014, pp. 1930–1936.

[158] R. Goel, S. M. Kazemi, M. Brubaker, and P. Poupart, "Diachronic embedding for temporal knowledge graph completion," in *AAAI*, 2020, pp. 3988–3995.

[159] R. Trivedi, H. Dai, Y. Wang, and L. Song, "Know-evolve: Deep temporal reasoning for dynamic knowledge graphs," in *ICML*, 2017, pp. 3462–3471.

[160] W. Jin, C. Zhang, P. Szekely, and X. Ren, "Recurrent event network for reasoning over temporal knowledge graphs," in *ICLR RLGM Workshop*, 2019.

[161] T. Jiang, T. Liu, T. Ge, L. Sha, B. Chang, S. Li, and Z. Sui, "Towards time-aware knowledge graph completion," in *COLING*, 2016, pp. 1715–1724.

[162] T. Jiang, T. Liu, T. Ge, L. Sha, S. Li, B. Chang, and Z. Sui, "Encoding temporal information for time-aware link prediction," in *EMNLP*, 2016, pp. 2350–2354.

[163] M. W. Chekol, G. Pirrò, J. Schoenfisch, and H. Stuckenschmidt, "Marrying uncertainty and time in knowledge graphs," in *AAAI*, 2017, pp. 88–94.

[164] Y.-N. Chen, W. Y. Wang, and A. Rudnicky, "Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding," in *NAACL*, 2015, pp. 619–629.

[165] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification." in *IJCAI*, 2017, pp. 2915–2921.

[166] H. Peng, J. Li, Q. Gong, Y. Song, Y. Ning, K. Lai, and P. S. Yu, "Fine-grained event categorization with heterogeneous graph convolutional networks," in *IJCAI*, 2019, pp. 3238–3245.

[167] R. Logan, N. F. Liu, M. E. Peters, M. Gardner, and S. Singh, "Barack's wife hillary: Using knowledge graphs for fact-aware language modeling," in *ACL*, 2019, pp. 5962–5971.

[168] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *ACL*, 2019, pp. 1441–1451.

[169] B. He, D. Zhou, J. Xiao, Q. Liu, N. J. Yuan, T. Xu *et al.*, "Integrating graph contextualized knowledge into pre-trained language models," *arXiv preprint arXiv:1912.00147*, 2019.

[170] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-BERT: Enabling language representation with knowledge graph," in *AAAI*, 2020, pp. 1–8.

[171] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, "ERNIE: Enhanced representation through knowledge integration," *arXiv preprint arXiv:1904.09223*, 2019.

[172] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "ERNIE 2.0: A continual pre-training framework for language understanding," in *AAAI*, 2020, pp. 8968–8975.

[173] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" in *EMNLP-IJCNLP*, 2019, pp. 2463–2473.

[174] A. Bordes, N. Usunier, S. Chopra, and J. Weston, "Large-scale simple question answering with memory networks," *arXiv preprint arXiv:1506.02075*, 2015.

[175] Z. Dai, L. Li, and W. Xu, "CFO: Conditional focused neural question answering with large-scale knowledge bases," in *ACL*, vol. 1, 2016, pp. 800–810.

[176] S. He, C. Liu, K. Liu, and J. Zhao, "Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning," in *ACL*, 2017, pp. 199–208.

[177] Y. Chen, L. Wu, and M. J. Zaki, "Bidirectional attentive memory networks for question answering over knowledge bases," in *NAACL*, 2019, pp. 2913–2923.

[178] S. Mohammed, P. Shi, and J. Lin, "Strong baselines for simple question answering over knowledge graphs with and without neural networks," in *NAACL*, 2018, pp. 291–296.

[179] L. Bauer, Y. Wang, and M. Bansal, "Commonsense for generative multi-hop question answering tasks," in *EMNLP*, 2018, pp. 4220–4230.

[180] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song, "Variational reasoning for question answering with knowledge graph," in *AAAI*, 2018, pp. 6069–6076.

[181] B. Y. Lin, X. Chen, J. Chen, and X. Ren, "KagNet: Knowledge-aware graph networks for commonsense reasoning," in *EMNLP-IJCNLP*, 2019, pp. 2829–2839.

[182] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, "Cognitive graph for multi-hop reading comprehension at scale," in *ACL*, 2019, pp. 2694–2703.

[183] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *SIGKDD*, 2016, pp. 353–362.

[184] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," in *WWW*, 2018, pp. 1835–1844.

[185] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in *WWW*, 2019, pp. 2000–2010.

[186] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *AAAI*, vol. 33, 2019, pp. 5329–5336.

[187] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, and Y. Zhang, "Reinforcement knowledge graph reasoning for explainable recommendation," in *SIGIR*, 2019.

[188] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *SIGKDD*, 2019, pp. 950–958.

[189] K. Hayashi and M. Shimbo, "On the equivalence of holographic and complex embeddings for link prediction," in *ACL*, 2017, pp. 554–559.

[190] A. Sharma, P. Talukdar *et al.*, "Towards understanding the geometry of knowledge graph embeddings," in *ACL*, 2018, pp. 122–131.

[191] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[192] M. Fan, Q. Zhou, E. Chang, and T. F. Zheng, "Transition-based knowledge graph embedding with relational mapping properties," in *PACLIC*, 2014, pp. 328–337.

[193] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *AAAI*, 2016, pp. 985–991.

[194] A. García-Durán, A. Bordes, and N. Usunier, "Effective blending of two and three-way interactions for modeling multi-relational data," in *ECML*. Springer, 2014, pp. 434–449.

[195] R. Reiter, "Deductive question-answering on relational data bases," in *Logic and data bases*. Springer, 1978, pp. 149–177.

[196] L. Cai and W. Y. Wang, "KBGAN: Adversarial learning for knowledge graph embeddings," in *NAACL*, 2018, pp. 1470–1480.

[197] D. Seyler, M. Yahya, and K. Berberich, "Knowledge questions from knowledge graphs," in *SIGIR*, 2017, pp. 11–18.

[198] C. Xiong, R. Power, and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *WWW*, 2017, pp. 1271–1279.

[199] C. Y. Li, X. Liang, Z. Hu, and E. P. Xing, "Knowledge-driven encode, retrieve, paraphrase for medical image report generation," *arXiv preprint arXiv:1903.10122*, 2019.

[200] M. Gaur, A. Alambo, J. P. Sain, U. Kursuncu, K. Thirunarayan, R. Kavuluru, A. Sheth, R. S. Welton, and J. Pathak, "Knowledge-aware assessment of severity of suicide risk for early intervention," in *WWW*, 2019, pp. 514–525.

[201] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *CVPR*, 2018, pp. 6857–6866.

[202] L. Liu, T. Zhou, G. Long, J. Jiang, and C. Zhang, "Attribute propagation network for graph zero-shot learning," in *AAAI*, 2020, pp. 4868–4875.

[203] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi, "Text generation from knowledge graphs with graph transformers," in *NAACL*, 2019, pp. 2284–2293.

[204] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *AAAI*, 2018, pp. 1795–1802.

[205] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm," in *AAAI*, 2018, pp. 5876–5883.

[206] Z. Liu, Z.-Y. Niu, H. Wu, and H. Wang, "Knowledge aware conversation generation with explainable reasoning over augmented graphs," in *EMNLP*, 2019, pp. 1782–1792.

[207] S. Moon, P. Shah, A. Kumar, and R. Subba, "OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs," in *ACL*, 2019, pp. 845–854.

[208] D. Guo, D. Tang, N. Duan, M. Zhou, and J. Yin, "Dialog-to-Action: Conversational question answering over a large-scale knowledge base," in *NeurIPS*, 2018, pp. 2942–2951.

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

[210] C. Matuszek, M. Witbrock, J. Cabral, and J. DeOliveira, "An introduction to the syntax and content of cyc," in *AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 2006, pp. 1–6.

[211] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *WWW*, 2007, pp. 697–706.

[212] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*, 2007, pp. 722–735.

[213] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *SIGMOD*, 2008, pp. 1247–1250.

[214] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *AAAI*, 2010, pp. 1306–1313.

[215] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledge base," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[216] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *ACL Workshop on CVSC*, 2015, pp. 57–66.

[217] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in *SIGMOD*, 2012, pp. 481–492.

[218] A. T. McCray, "An upper-level ontology for the biomedical domain," *International Journal of Genomics*, vol. 4, no. 1, pp. 80–84, 2003.

[219] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *ECML*, 2010, pp. 148–163.

[220] E. Cambria, Y. Song, H. Wang, and N. Howard, "Semantic multidimensional scaling for open-domain sentiment analysis," *IEEE Intelligent Systems*, vol. 29, no. 2, pp. 44–51, 2012.

[221] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, and M. Sun, "Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation," in *EMNLP*, 2018, pp. 4803–4809.

[222] J. Hao, M. Chen, W. Yu, Y. Sun, and W. Wang, "Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts," in *KDD*, 2019, pp. 1709–1719.

[223] S. Ahn, H. Choi, T. Pärnamaa, and Y. Bengio, "A neural knowledge language model," *arXiv preprint arXiv:1608.00318*, 2016.

[224] P. Trivedi, G. Maheshwari, M. Dubey, and J. Lehmann, "LC-QuAD: A corpus for complex question answering over knowledge graphs," in *ISWC*, 2017, pp. 210–218.

[225] L. Costabello, S. Pai, C. L. Van, R. McGrath, and N. McCarthy, "AmpliGraph: a Library for Representation Learning on Knowledge Graphs," 2019.

[226] X. Han, S. Cao, L. Xin, Y. Lin, Z. Liu, M. Sun, and J. Li, "OpenKE: An open toolkit for knowledge embedding," in *EMNLP*, 2018, pp. 139–144.

[227] X. Han, T. Gao, Y. Yao, D. Ye, Z. Liu, and M. Sun, "OpenNRE: An open and extensible toolkit for neural relation extraction," in *EMNLP-IJCNLP*, 2019, pp. 169–174.