A Survey on Knowledge Graphs: Representation, Acquisition, and Applications

Shaoxiong Ji, Shirui Pan, *Member, IEEE*, Erik Cambria, *Fellow, IEEE*, Pekka Marttinen, and 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 toward cognition and human-level intelligence. In this survey, we provide a comprehensive review of the 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 metarelational learning, commonsense reasoning, and temporal knowledge graphs. To facilitate future research on knowledge graphs, we also provide a curated collection of data sets and open-source libraries on different tasks. In the end, we have a thorough outlook on several promising research directions.**

***Index Terms*—Deep learning, knowledge graph completion (KGC), knowledge graph, reasoning, relation extraction, representation learning.**

NOMENCLATURE

$\mathcal{G}$	Knowledge graph.
$\mathcal{F}$	Set of facts.
(h, r, t)	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 the vertex set.
$\xi \in \mathcal{E}_\mathcal{G}$	Edge in the 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.
$\sigma(\cdot)$, $g(\cdot)$	Nonlinear 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 (elementwise) product.
$\mathbf{h} \star \mathbf{t}$	Circular correlation.
concat(), $[\mathbf{h}, \mathbf{r}]$	Vectors/matrices concatenation.
$\boldsymbol{\omega}$	Convolutional filters.
$*$	Convolution operator.

Manuscript received August 9, 2020; revised January 16, 2021; accepted March 30, 2021. Date of publication April 26, 2021; date of current version February 4, 2022. This work was supported in part by NSF under Grant III-1763325, Grant III-1909323, and Grant SaTC-1930941; in part by the Agency for Science, Technology and Research (A*STAR) through its AME Programmatic Funding Scheme under Project A18A2b0046; in part by the Academy of Finland under Grant 336033 and Grant 315896; in part by Business Finland under Grant 884/31/2018; and in part by EU H2020 under Grant 101016775. *(Corresponding author: Shirui Pan.)*

Shaoxiong Ji and Pekka Marttinen are with the Department of Computer Science, Aalto University, 02150 Aalto, Finland (e-mail: shaoxiong.ji@aalto.fi; pekka.marttinen@aalto.fi).

Shirui Pan is with the Department of Data Science and AI, Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia (e-mail: shirui.pan@monash.edu).

Erik Cambria is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: cambria@ntu.edu.sg).

Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: psyu@uic.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3070843

Digital Object Identifier 10.1109/TNNLS.2021.3070843

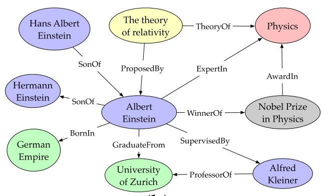
I. INTRODUCTION

INCORPORATING human knowledge is one of the research directions of artificial intelligence (AI). Knowledge representation and reasoning, inspired by human problem solving, are 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 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,

(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)



(b)

Fig. 1. Example of knowledge base and knowledge graph. (a) Factual triples in knowledge base. (b) Entities and relations in knowledge graph.

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 article uses the terms knowledge graph and knowledge base interchangeably.

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 multilingual 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 article 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.

1) *Comprehensive Review:* We conduct a comprehensive review of the origin of knowledge graphs and modern techniques for relational learning on knowledge graphs. 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.
2) *Full-View Categorization and New Taxonomies:* A full-view categorization of research on knowledge graph, together with fine-grained new taxonomies, is presented. Specifically, at the high level, we review the research on knowledge graphs in four aspects: KRL, knowledge acquisition, temporal knowledge graphs, and knowledge-aware applications. For KRL, 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 metarelational learning; entity acquisition tasks are divided into entity recognition, typing, disambiguation, and alignment; and relation extraction is discussed according to the neural paradigms.
3) *Wide Coverage on Emerging Advances:* We provide wide coverage on emerging topics, including transformer-based knowledge encoding, graph neural network (GNN)-based knowledge propagation, reinforcement learning (RL)-based path reasoning, and metarelational learning.
4) *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 Sections IV and 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 data sets and open-source implementations, can be found in the appendixes.

II. Overview

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 first 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 first 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. 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. 1 in Appendix A in the Supplementary Material. Many general knowledge graph databases and domain-specific knowledge bases have been released to facilitate research. We introduce more general and domain-specific knowledge bases in Appendixes F-A1 and F-A2 in the Supplementary Material.

[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

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 multirelational 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 multirelational 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 Nomenclature. Details of several mathematical operations are explained in Appendix B in the Supplementary Material.

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. 2.

Knowledge Representation Learning is a critical research issue of the 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*, providing a clear workflow for developing a KRL model. Specific ingredients include the following:

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 pointwise space, manifold, complex vector space, the Gaussian distribution, and discrete space. Scoring metrics are generally divided into the distance- 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 (also known as relations and entities) from the text. KGC falls into the following categories: embedding-based ranking, relation path reasoning, rule-based reasoning, and metarelational learning. Entity discovery includes recognition, disambiguation, typing, and alignment. Relation extraction models utilize attention mechanisms, graph convolutional networks (GCNs), adversarial training (AT), RL, deep residual learning, and transfer learning.

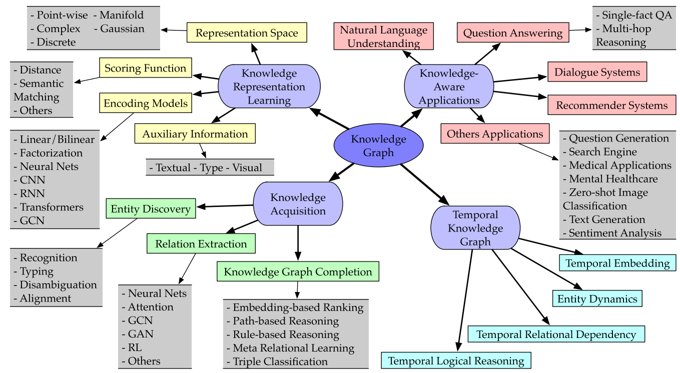


Fig. 2. Categorization of research on knowledge graphs.

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

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 into the flow of KRL and provides a full-scaled view from fourfold, 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, multirelation 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 D in the Supplementary Material.

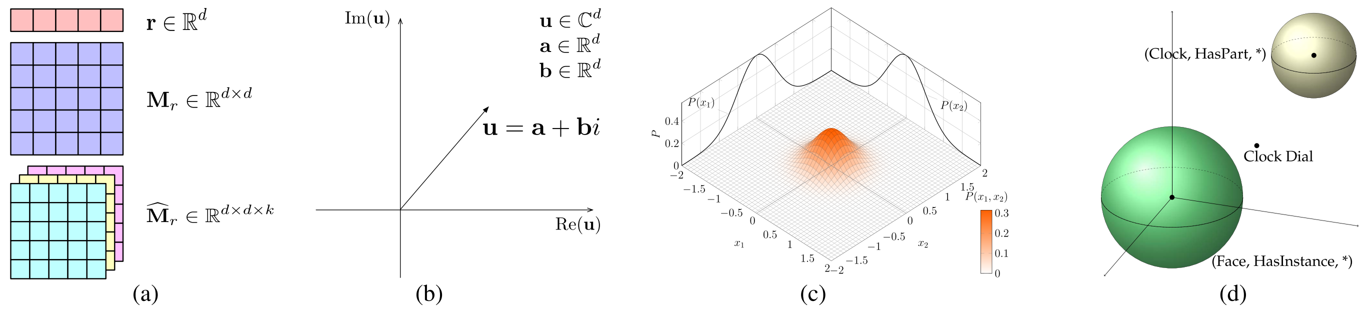


Fig. 3. Illustration of knowledge representation in different spaces. (a) Pointwise space. (b) Complex vector space. (c) Gaussian distribution. (d) Manifold space.

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 pointwise space [see Fig. 3(a)], including vector, matrix, and tensor space, while other kinds of space, such as complex vector space [see Fig. 3(b)], Gaussian space [see Fig. 3(c)], and manifold [see Fig. 3(d)], are utilized as well. The embedding space should follow three conditions, i.e., differentiability, calculation possibility, and definability of a scoring function [15].

1) Pointwise Space: The pointwise Euclidean space is widely applied for representing entities and relations, projecting relation embedding in vector or matrix space, or capturing relational interactions. TransE [16] 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 [17] 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 [18] 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 [19] 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 [20], also use similar representation space, while semantic matching models use plain vector space (e.g., HolE [21]) and relational projection matrix (e.g., ANALOGY [22]). Principles of these translational and semantic matching models are introduced in Sections 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 [23] first introduces complex vector space shown in Fig. 3(d), which can capture both symmetric and antisymmetric relations. The Hermitian dot product is used to do composition for relation, head, and the conjugate of the tail. Inspired by Euler's identity $e^{i\theta} = \cos\theta + i\sin\theta$, RotatE [24] 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 elementwise Hadmard product. QuatE [25] 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. With the introduction of the rotational Hadmard product in complex space, RotatE [24] can also capture inversion and composition patterns, as well as symmetry and antisymmetry. QuatE [25] uses the Hamilton product to capture latent interdependencies within the 4-D space of entities and relations and gains a more expressive rotational capability than RotatE.

3) Gaussian Distribution: Inspired by the Gaussian word embedding, the density-based embedding model KG2E [26] introduces Gaussian distribution to deal with the (un)certainties of entities and relations. The authors embedded entities and relations into multidimensional 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 [27] represents entities with Gaussian distributions, while it draws a mixture of Gaussian distribution for relation embedding, where the mth component translation vector of relation r is denoted as $\mathbf{u}_{r,m} = \mathbf{t} - \mathbf{h} \sim \mathcal{N}(\mathbf{u_t} - \mathbf{u_h}, (\sigma_h^2 + \sigma_t^2)\mathbf{E})$.

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 pointwise 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 [28] extends pointwise 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. 3(d). For the sphere setting, reproducing kernel Hilbert space is used to represent the manifold function. Another "hyperplane" setting is introduced to enhance the model with intersected embeddings. ManifoldE [28] relaxes the real-valued pointwise space into manifold space with a more expressive representation from the geometric perspective. When the manifold function and relation-specific manifold parameter are set to zero, the manifold collapses into a point.

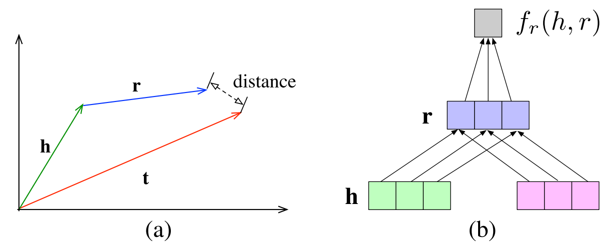


Fig. 4. Illustrations of distance-based and similarity matching-based scoring functions taking TransE [16] and DistMult [32] as examples. (a) Translational distance-based scoring of TransE. (b) Semantic similarity-based scoring of DistMult.

Hyperbolic space, a multidimensional Riemannian manifold with a constant negative curvature $-c$ $(c > 0)$: $\mathbb{B}^{d,c} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|^2 < (1/c)\}$, is drawing attention for its capacity of capturing hierarchical information. MuRP [29] represents the multirelational knowledge graph in the Poincaré ball of hyperbolic space $\mathbb{B}_c^d = \{\mathbf{x} \in \mathbb{R}^d : c\|\mathbf{x}\|^2 < 1\}$ while it fails to capture logical patterns and suffers from constant curvature. Chami *et al.* [30] leverages expressive hyperbolic isometries and learns a relation-specific absolute curvature c_r in the hyperbolic space.

TorusE [15] 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-D polygon. It utilizes a 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.

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 that positive samples have higher scores than negative samples. In this article, the term of the scoring function is adopted for unification. There are two typical types of scoring functions, i.e., distance- [see Fig. 4(a)] and similarity-based [see Fig. 4(b)] functions, to measure the plausibility of a fact. The 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 a 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.

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

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

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.* [16] 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 defined under L_1 or L_2 constraints as

$$f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_1/L_2}. \tag{2}$$

Since that, many variants and extensions of TransE have been proposed. For example, TransH [20] projects entities and relations into a hyperplane, TransR [17] 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 the Euclidean distance, TransA [34] uses the 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 the scoring function defined as

$$f_r(h,t) = \left\|\boldsymbol{\alpha}_r^H \cdot \mathbf{D} \cdot \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_r^T \cdot \mathbf{D} \cdot \mathbf{t}\right\|_\ell \tag{3}$$

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

$$f_r(\mathbf{h},\mathbf{t}) = \| - \tanh(\mathbf{t}\circ\mathbf{r})\circ\mathbf{h} + \mathbf{r} - \tanh(\mathbf{h}\circ\mathbf{r})\circ\mathbf{t} + \alpha\cdot(\mathbf{h}\circ\mathbf{t})\|_{\ell_{1/2}}. \tag{4}$$

KG2E [26] in the Gaussian space and ManifoldE [28] 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

$$f_r(h,t) = \left\|\mathcal{M}(h,r,t) - D_r^2\right\|^2 \tag{5}$$

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 blocks—i.e.,

$$f_r(h,t) = g_{\text{left}}(\mathbf{h},\mathbf{r})^\top g_{\text{right}}(\mathbf{r},\mathbf{t}). \tag{6}$$

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 multirelational representation learning, DistMult [32] proposes a simplified bilinear formulation defined as

$$f_r(h,t) = \mathbf{h}^\top \operatorname{diag}(\mathbf{M}_r)\mathbf{t}. \tag{7}$$

To capture productive interactions in relational data and compute efficiently, HolE [21] introduces a circular correlation

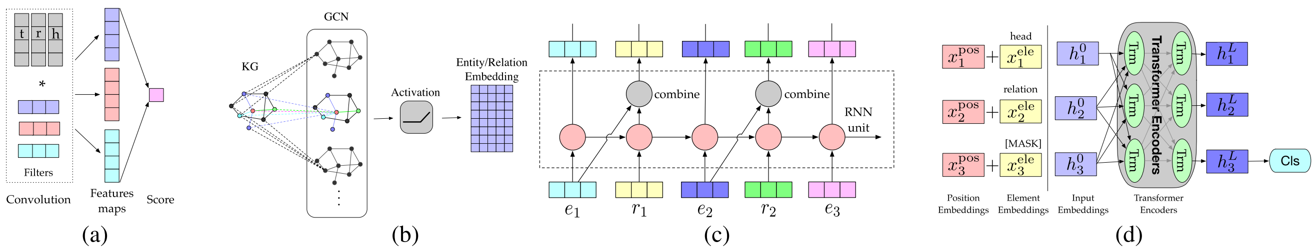


Fig. 5. Illustrations of neural encoding models. (a) CNN [43] input triples into dense layer and convolution operation to learn semantic representation. (b) GCN [44] acts as encoder of knowledge graphs to produce entity and relation embeddings. (c) RSN [45] encodes entity–relation sequences and skips relations discriminatively. (d) Transformer-based CoKE [46] encodes triples as sequences with an entity replaced by `[MASK]`.

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 the linear concatenation of perturbed HolE. Focusing on multirelational inference, ANALOGY [22] models analogical structures of relational data. Its scoring function is defined as

$$f_r(h, t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t} \tag{8}$$

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. HolE with Fourier transformed in the frequency domain can be viewed as a special case of ComplEx [41], which connects holographic and complex embeddings. The analogical embedding framework [22] can recover or equivalently obtain several models, such as DistMult, ComplEx, and HolE, by restricting the embedding dimension and scoring function. Crossover interactions are introduced by CrossE [42] with an interaction matrix $\mathbf{C} \in \mathbb{R}^{n_r \times d}$ to simulate the bidirectional 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

$$f(h, r, t) = \sigma\big(\tanh(\mathbf{c}_r \circ \mathbf{h} + \mathbf{c}_r \circ \mathbf{h} \circ \mathbf{r} + \mathbf{b})\mathbf{t}^\top\big). \tag{9}$$

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

The two methods mentioned above in Section III-A4 with group representation also follow the semantic matching principle. The scoring function of TorusE [15] is defined as

$$\min_{(x,y)\in([h]+[r])\times[t]} \|x - y\|_i. \tag{10}$$

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

$$f_r(h, t) = \mathbf{h}^\top \mathbf{R} \mathbf{t} = \sum_{l=1}^{L} \mathbf{h}^{(l)\top} \mathbf{R}^{(l)} \mathbf{t}^{(l)} \tag{11}$$

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 nonlinear neural activation and more complex network structures by matching semantic similarity of entities and relations. Several neural models are illustrated in Fig. 5.

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

$$g_r(\mathbf{h}, \mathbf{t}) = \mathbf{M}_r^T \begin{pmatrix} \mathbf{h} \\ \mathbf{t} \end{pmatrix} \tag{12}$$

or bilinear transformation operations as (8). Canonical methods with linear/bilinear encoding include SE [8], SME [39], DistMult [32], ComplEx [23], and ANALOGY [22]. For TransE [16] with L2 regularization, the scoring function can be expanded to the form with only linear transformation with 1-D vectors, i.e.,

$$\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2 = 2\mathbf{r}^T(\mathbf{h}-\mathbf{t}) - 2\mathbf{h}^T\mathbf{t} + \|\mathbf{r}\|_2^2 + \|\mathbf{h}\|_2^2 + \|\mathbf{t}\|_2^2. \tag{13}$$

Wang *et al.* [47] 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 [48] introduces the inverse of relations and calculates the average canonical Polyadia score of (h, r, t) and (t, r^{-1}, h) as

$$f_r(h, t) = \frac{1}{2}\big(\mathbf{h} \circ \mathbf{r}\mathbf{t} + \mathbf{t} \circ \mathbf{r}'\mathbf{t}\big) \tag{14}$$

where $\mathbf{r}'$ is the embedding of inversion relation. Embedding models in the bilinear family, such as RESCAL, DistMult, HolE, and ComplEx, can be transformed from one into another with certain constraints [47]. More bilinear models are proposed from a factorization perspective discussed in Section III-C2.

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.* [49] proposed the three-way rank-r factorization RESCAL over each relational slice of knowledge graph tensor. For the kth relation of m relations, the kth slice of $\mathcal{X}$ is factorized as

$$\mathcal{X}_k \approx \mathbf{A}\mathbf{R}_k\mathbf{A}^T. \tag{15}$$

The authors further extended it to handle attributes of entities efficiently [50]. Jenatton *et al.* [51] 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 [52] learns to embed by outputting a core tensor and embedding vectors of entities and relations. LowFER [53] 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 the multilayer perceptron (MLP) [3], the neural tensor network (NTN) [18], and the neural association model (NAM) [54]. They generally feed entities or relations or both into deep neural networks and compute a semantic matching score. MLP [3] encodes entities and relations together into a fully connected layer and uses a second layer with sigmoid activation for scoring a triple as

$$f_r(h,t) = \sigma(\mathbf{w}^\top \sigma(\mathbf{W}[\mathbf{h},\mathbf{r},\mathbf{t}])) \tag{16}$$

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 [18] takes entity embeddings as input associated with a relational tensor and outputs predictive score as

$$f_r(h,t) = \mathbf{r}^\top \sigma(\mathbf{h}^T \widehat{\mathbf{M}}\mathbf{t} + \mathbf{M}_{r,1}\mathbf{h} + \mathbf{M}_{r,2}\mathbf{t} + \mathbf{b}_r) \tag{17}$$

where $\mathbf{b}_r \in \mathbb{R}^k$ is bias for relation r, and $\mathbf{M}_{r,1}$ and $\mathbf{M}_{r,2}$ are relation-specific weight matrices. It can be regarded as a combination of MLPs and bilinear models. NAM [54] associates the hidden encoding with the embedding of the tail entity and proposes the relational-modulated neural network (RMNN).

4) Convolutional Neural Networks: CNNs are utilized for learning deep expressive features. ConvE [55] uses 2-D convolution over embeddings and multiple layers of nonlinear features to model the interactions between entities and relations by reshaping head entity and relation into 2-D 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

$$f_r(h,t) = \sigma(\mathrm{vec}(\sigma([\mathbf{M}_h;\mathbf{M}_r] * \boldsymbol{\omega}))\mathbf{W})\mathbf{t} \tag{18}$$

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 nonlinear feature learning through multiple layers. ConvKB [43] adopts CNNs for encoding the concatenation of entities and relations without reshaping [see Fig. 5(a)]. Its scoring function is defined as

$$f_r(h,t) = \mathrm{concat}(\sigma([\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}] * \boldsymbol{\omega})) \cdot \mathbf{w}. \tag{19}$$

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 [56] utilizes hypernetwork $\mathbf{H}$ for 1-D relation-specific convolutional filter generation to achieve multitask knowledge sharing and, meanwhile, simplifies 2-D 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 triplet-level representations. In comparison, recurrent networks can capture long-term relational dependencies in knowledge graphs. Gardner *et al.* [57] and Neelakantan *et al.* [58] propose the RNN-based model over the relation path to learn vector representation without and with entity information, respectively. RSN [45] [see Fig. 5(c)] 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

$$\mathbf{h}_t' = \begin{cases} \mathbf{h}_t, & x_t \in \mathcal{E} \\ \mathbf{S}_1\mathbf{h}_t + \mathbf{S}_2\mathbf{x}_{t-1}, & x_t \in \mathcal{R} \end{cases} \tag{20}$$

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 [46] employs transformers to encode edges and path sequences. Similarly, KG-BERT [59] borrows the idea from language model pre-training and takes the 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 [60] proposes relation-specific transformation to model the directed nature of knowledge graphs. Its forward propagation is defined as

$$x_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} x_j^{(l)} + W_0^{(l)} x_i^{(l)}\right) \tag{21}$$

where $x_i^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of the ith entity in the lth layer, N_i^r is a neighbor set of the ith 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 [61] 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 [44] introduces weighted GCN [see Fig. 5(b)], 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 the 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

$$f_r(h,t) = g(\mathrm{vec}(\mathbf{M}(\mathbf{h},\mathbf{r}))W)\mathbf{t}. \tag{22}$$

Nathani *et al.* [62] introduced graph attention networks with multihead attention as the encoder to capture multihop neighborhood features by inputting the concatenation of entity and relation embeddings. CompGCN [63] 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

Multimodal 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 $\mathcal{D} = \langle w_1, w_2, \ldots, w_n \rangle$, 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.* [64] proposed two alignment models for aligning entity space and word space by introducing entity names and Wikipedia anchors. DKRL [65] extends TransE [16] to learn representation directly from entity descriptions by a convolutional encoder. SSP [66] 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.* [64] used a three-component loss $\mathcal{L} = \mathcal{L}_K + \mathcal{L}_T + \mathcal{L}_A$ of the knowledge model $\mathcal{L}_K$, text model $\mathcal{L}_T$ and the alignment model $\mathcal{L}_A$. SSP [66] uses a two-component objective function $\mathcal{L} = \mathcal{L}_{\text{embed}} + \mu \mathcal{L}_{\text{topic}}$ of embedding-specific loss $\mathcal{L}_{\text{embed}}$ and topic-specific loss $\mathcal{L}_{\text{topic}}$ 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 [67] incorporates semantic categories of entities to embed entities belonging to the same category smoothly in semantic space. TKRL [68] proposes type encoder model for projection matrix of entities to capture type hierarchy. Noticing that some relations indicate attributes of entities, KR-EAR [69] categorizes relation types into attributes and relations and modeled the correlations between entity descriptions. Zhang *et al.* [70] extended existing embedding methods with hierarchical relation structure of relation clusters, relations, and subrelations.

3) Visual Information: Visual information (e.g., entity images) can be utilized to enrich KRL. Image-embodied IKRL [71], 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- 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 using additional information. This article discusses relation path and logical rules under the umbrella of KGC in Sections IV-A2 and IV-A4, respectively.

4) Uncertain Information: Knowledge graphs, such as ProBase [72], NELL [73], and ConceptNet [74], contain uncertain information with a confidence score assigned to every relational fact. In contrast to classic deterministic KGE, uncertain embedding models aim to capture uncertainty representing the likelihood of relational facts. Chen *et al.* [75] proposed an uncertain KGE model to simultaneously preserve structural and uncertainty information, where probabilistic soft logic is applied to infer the confidence score. Probability calibration takes a postprocessing process to adjust probability scores, making predictions probabilistic sense. Tabacof and Costabello [76] first studied probability calibration for KGE under the closed-world assumption, revealing that well-calibrated models can lead to improved accuracy. Safavi *et al.* [77] further explored probability calibration under the more challenging open-world assumption.

E. Summary

KRL is vital in the research community of knowledge graphs. This section reviews four folds of KRL with several modern methods summarized in Table I and more in Appendix C in the Supplementary Material. 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 triplets in a specific space; 3) which encoding model to use for modeling relational interactions; and 4) whether to utilize auxiliary information. The most popularly used representation space is the 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 pointwise Euclidean space by relaxing the pointwise 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, entity images, and confidence scores.

IV. Knowledge Acquisition

Knowledge acquisition aims to construct knowledge graphs from unstructured text and other structured or semistructured 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 (EA). Most methods formulate KGC and relation extraction separately. These two tasks, however, can also be integrated into a unified framework. Han *et al.* [78] 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

TABLE I

SUMMARY OF RECENT KRL MODELS. SEE MORE DETAILS IN APPENDIX C IN THE SUPPLEMENTARY MATERIAL

Model	Ent. & Rel. embed.	Scoring Function $f_r(h,t)$
RotatE [24]	$\mathbf{h},\mathbf{t} \in \mathbb{C}^d$, $\mathbf{r} \in \mathbb{C}^d$	$\lVert \mathbf{h} \circ \mathbf{r} - \mathbf{t} \rVert$
TorusE [15]	$[\mathbf{h}],[\mathbf{t}] \in \mathbb{T}^n$, $[\mathbf{r}] \in \mathbb{T}^n$	$\min_{(x,y)\in([h]+[r])\times[t]} \lVert x-y \rVert_i$
SimplE [48]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r},\mathbf{r}' \in \mathbb{R}^d$	$\frac{1}{2}\left(\mathbf{h}\circ\mathbf{r}\mathbf{t}+\mathbf{t}\circ\mathbf{r}'\mathbf{t}\right)$
TuckER [52]	$\mathbf{h},\mathbf{t} \in \mathbb{R}_e^d$, $\mathbf{r} \in \mathbb{R}_r^d$	$\mathcal{W} \times_1 \mathbf{h} \times_2 \mathbf{r} \times_3 \mathbf{t}$
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 [42]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\sigma\left(\sigma\left(\mathbf{c}_r\circ\mathbf{h}+\mathbf{c}_r\circ\mathbf{h}\circ\mathbf{r}+\mathbf{b}\right)\mathbf{t}^\top\right)$
QuatE [25]	$\mathbf{h},\mathbf{t} \in \mathbb{H}^d$, $\mathbf{r} \in \mathbb{H}^d$	$\mathbf{h}\otimes\frac{\mathbf{r}}{\lvert\mathbf{r}\rvert}\cdot\mathbf{t}$
SACN [44]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$g\left(\mathrm{vec}\left(\mathbf{M}\left(\mathbf{h},\mathbf{r}\right)\right)W\right)\mathbf{t}$
ConvKB [43]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\mathrm{concat}\left(g\left(\left[\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\right]*\omega\right)\right)\mathbf{w}$
ConvE [55]	$\mathbf{M}_h \in \mathbb{R}^{d_w\times d_h}$, $\mathbf{t}\in\mathbb{R}^d$ $\mathbf{M}_r \in \mathbb{R}^{d_w\times d_h}$	$\sigma\left(\mathrm{vec}\left(\sigma\left(\left[\mathbf{M}_h;\mathbf{M}_r\right]*\boldsymbol{\omega}\right)\right)\mathbf{W}\right)\mathbf{t}$
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 [19]	$\mathbf{h}_m,\mathbf{t}_m \in \mathbb{R}^d, \mathbf{r}_m \in \mathbb{R}_+^d$ $\mathbf{h}_p,\mathbf{r}_p,\mathbf{t}_p \in [0,2\pi)^d$	$-\lVert\mathbf{h}_m\circ\mathbf{r}_m-\mathbf{t}_m\rVert_2-$ $\lambda\left\lVert\sin\left(\left(\mathbf{h}_p+\mathbf{r}_p-\mathbf{t}_p\right)/2\right)\right\rVert_1$
MuRP [29]	$\mathbf{h},\mathbf{t},\mathbf{r} \in \mathbb{B}_c^d, b_h, b_t \in \mathbb{R}$	$-d_{\mathbb{B}}\left(\mathbf{h}^{(r)},\mathbf{t}^{(r)}\right)^2+b_s+b_o$
AttH [30]	$\mathbf{h},\mathbf{t},\mathbf{r} \in \mathbb{B}_c^d, b_h, b_t \in \mathbb{R}$	$-d_{\mathbb{B}}^{c_r}\left(Q(h,r),\mathbf{e}_t^H\right)^2+b_h+b_t$
LowFER [53]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d, \mathbf{r} \in \mathbb{R}^d$	$\left(\mathbf{S}^k\,\mathrm{diag}\left(\mathbf{U}^T\mathbf{h}\right)\mathbf{V}^T\mathbf{r}\right)^T\mathbf{t}$

acquisition, such as triple classification [79], relation classification [80], and open knowledge enrichment [81]. In this section, three categories of knowledge acquisition techniques, namely, 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 multistep relationships. Thus, recent work turns to explore multistep relation paths and incorporate logical rules, termed *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. 6(a), first 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 [16], TransH [20], TransR [17], HolE [21], and R-GCN [60]) and joint learning methods, such as DKRL [65] with textual information, can been used for KGC.

Unlike representing inputs and candidates in the unified embedding space, ProjE [82] 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 an output projection

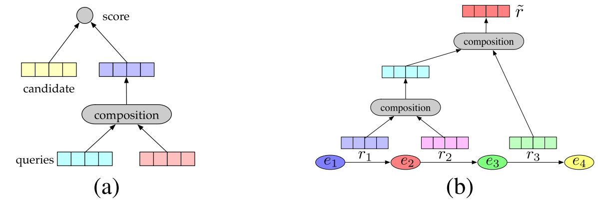


Fig. 6. (a) Embedding-based ranking and (b) relation path reasoning [58].

layer, is defined as $h(\mathbf{e},\mathbf{r}) = g(\mathbf{W}^c\sigma(\mathbf{e}\oplus\mathbf{r})+b_p)$, 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 [83] 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 [84] 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 [85] proposes a generative model, called conditional relationship variational autoencoder, for entity pair discovery from latent space.

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) [86] chooses a relational path under a combination of path constraints and conducts maximum-likelihood classification. To improve path search, Gardner *et al.* [57] introduced vector space similarity heuristics in the random walk by incorporating textual content, which also relieves the feature sparsity issue in PRA. Neural multihop relational path modeling is also studied. Neelakantan *et al.* [58] developed an RNN model to compose the implications of relational paths by applying compositionality recursively [in Fig. 6(b)]. Chain-of-Reasoning [87], a neural attention mechanism to enable multiple reasons, represents logical composition across all relations, entities, and text. Recently, DIVA [88] proposes a unified variational inference framework that takes multihop reasoning as two substeps of path-finding (a prior distribution for underlying path inference) and path-reasoning (a likelihood for link classification).

3) RL-Based Path Finding: Deep RL is introduced for multihop 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 [89] first applies RL into relational path learning and develops a novel reward function to improve accuracy, path diversity, and path efficiency. It encodes states in

TABLE II

COMPARISON OF RL-BASED PATH FINDING FOR KNOWLEDGE GRAPH REASONING

Method	State s_t	Action a_t	Reward γ	Policy Network
DeepPath [89]	$(\mathbf{e}_t, \mathbf{e}_q - \mathbf{e}_t)$	$\{r\}$	Global 1 $\mathrm{e_t} = \mathrm{e_q}$ or -1 $\mathrm{e_t} \neq \mathrm{e_q}$ Efficiency $\frac{1}{\mathtt{length}(p)}$ Diversity $-\frac{1}{\lvert F\rvert}\sum_{i=1}^{\lvert F\rvert}\cos(\mathbf{p}, \mathbf{p}_i)$	Fully-connected network (FCN)
MINERVA [90]	(e_t, e_s, r_q, e_q)	$\{(e_t, r, v)\}$	$\mathbb{I}\{\mathrm{e_t} = \mathrm{e_q}\}$	$\mathbf{h_t} = LSTM\left(\mathbf{h_{t-1}}, [\mathbf{a_{t-1}}; \mathbf{o_t}]\right)$
Multi-Hop [91]	$(e_t, (e_s, r_q))$	$\left\{(r', e') \mid (e_t, r', e') \in \mathcal{G}\right\}$	$\gamma + (1-\gamma) f_{r_q}(e_s, e_T)$	$\mathbf{h}_t = LSTM\left(\mathbf{h}_{t-1}, \mathbf{a}_{t-1}\right)$
M-Walk [92]	$s_{t-1} \cup \left\{a_{t-1}, v_t, \mathcal{E}_{\mathcal{G}_{v_t}}, \mathcal{V}_{v_t}\right\}$	$\bigcup_t \mathcal{E}_{\mathcal{G}_{v_t}} \cup \{\mathrm{STOP}\}$	$\mathbb{I}\{\mathrm{e_t} = \mathrm{e_q}\}$	GRU-RNN + FCN
CPL [93] Reasoner	(e_s, r_q, h_t)	$\{\xi \in \mathcal{E}_{\mathcal{G}}\}$	$\mathbb{I}\{\mathrm{e_t} = \mathrm{e_q}\}$	$\mathbf{h}_t = LSTM\left(\mathbf{h}_{t-1}, [\mathbf{r}_t, \mathbf{e}_t]\right)$
CPL [93] Extractor	(b_{e_t}, e_t)	$\left\{(r', e')\right\}_{(e_t, r', e')} \in b_{e_t}$	step-wise delayed from reasoner	PCNN-ATT

the continuous space via a translational embedding method and takes the relation space as its action space. Similarly, MINERVA [90] 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 [91] 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 [92] 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 [93] 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 II, where MINERVA, M-Walk, and CPL use the binary reward. For the policy networks, DeepPath uses a fully connected network, and 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

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

Logical rules can be extracted by rule mining tools, such as AMIE [94]. The recent RLvLR [95] 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 [96] 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. 7(a) illustrates a simple first-order Horn clause inference. RUGE [97] proposes an iterative model, where soft rules are utilized for soft label prediction from unlabeled triples and labeled triples for embedding rectification. IterE [98] proposes an iterative training strategy with three components of embedding learning, axiom induction, and axiom injection.

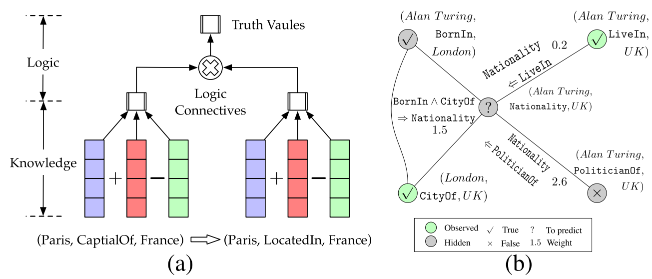


Fig. 7. Illustrations of logical rule learning. (a) KALE [96]. (b) pLogicNet [102].

The logical rule is one kind of auxiliary information; meanwhile, it can incorporate prior knowledge, enabling the ability of interpretable multihop reasoning and paving the way for generalization even in few-shot labeled relational triples. However, logic rules alone can only cover a limited number of relational facts in knowledge graphs and suffer colossal search space. The combination of neural and symbolic computation has complementary advantages that utilize efficient data-driven learning and differentiable optimization and exploit prior logical knowledge for precise and interpretable inference. Incorporating rule-based learning for knowledge representation is principally to add regularizations or constraints to representations. The neural theorem prover (NTP) [99] learns logical rules for multihop reasoning, which utilizes a radial basis function kernel for differentiable computation on the vector space. NeuralLP [100] enables gradient-based optimization to be applicable in inductive logic programming, where a neural controller system is proposed by integrating attention mechanism and auxiliary memory. Neural-Num-LP [101] extends NeuralLP to learn numerical rules with dynamic programming and cumulative sum operations. pLogicNet [102] proposes probabilistic logic neural networks [see Fig. 7(b)] 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 [103] generalizes pLogicNet by tuning graph networks and embedding and achieves more efficient logical reasoning.

5) Metarelational 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 *metarelational learning* or *few-shot relational learning*, requires models to predict new relational facts with only a very few samples.

Targeting the previous two observations, GMatching [104] 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 multistep matching guided by long short-term memory (LSTM) networks to calculate the similarity scores. Meta-KGR [105], an optimization-based metalearning approach, adopts model agnostic metalearning for fast adaption and RL for entity searching and path reasoning. Inspired by model- and optimization-based metalearnings, MetaR [106] transfers relation-specific metainformation from support set to query set and archives fast adaption via loss gradient of high-order relational representation. Zhang *et al.* [107] proposed joint modules of heterogeneous graph encoder, recurrent autoencoder, and matching network to complete new relational facts with few-shot references. Qin *et al.* [108] utilized GAN to generate reasonable embeddings for unseen relations under the zero-shot learning setting. Baek *et al.* [109] proposed a transductive metalearning framework, called graph extrapolation networks (GENs), for a few-shot out-of-graph link prediction in knowledge graphs.

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, such as TransH [20] and TransR [17], and semantic matching-based methods, such as NTN [18], HolE [21], and ANALOGY [22].

Vanilla vector-based embedding methods failed to deal with 1-to-n relations. Recently, Dong *et al.* [79] 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 EA. 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. Handcrafted features, such as capitalization patterns and language-specific resources, such as gazetteers, are applied in many pieces of literature. Recent work applies sequence-to-sequence neural architectures, for example, LSTM-CNN [110] for learning character- and word-level features and encoding partial lexicon matches. Lample *et al.* [111] proposed stacked neural architectures by stacking LSTM layers and CRF layers, i.e., LSTM-CRF [in Fig. 8(a)] and Stack-LSTM. MGNER [112] proposes an integrated framework with entity position detection in various granularities and attention-based entity classification for both nested and nonoverlapping named entities. Hu *et al.* [113] distinguished multitoken and single-token entities with multitask training. Recently, Li *et al.* [114] formulated flat and nested NER as a unified machine reading comprehension framework by referring annotation guidelines to construct query questions. Pretrained language models with knowledge graphs, such as ERNIE [115] and K-BERT [116], have been applied into NER and achieved improved performance.

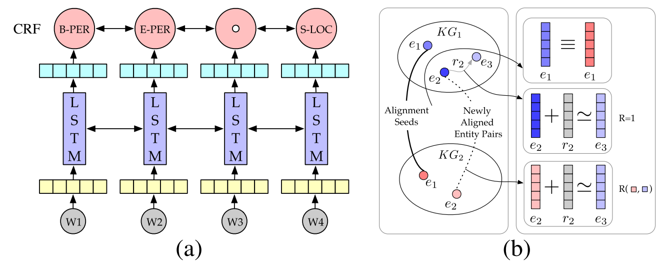


Fig. 8. Illustrations of several entity discovery tasks. (a) Entity recognition with LSTM-CRF [111]. (b) EA with IPTransE [126].

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 multiclass and multilabel classification. To reduce label noise, PLE [117] 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.* [118] proposed prototype-driven label embedding with hierarchical information for zero-shot fine-grained named entity typing. Recent studies utilize embedding-based approaches. For example, JOIE [119] learns joint embeddings of instance- and ontology-view graphs and formulates entity typing as top-k ranking to predict associated concepts. ConnectE [120] explores local typing and global triple knowledge to enhance joint embedding learning.

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 [121] for modeling entity semantic relatedness and EDKate [122] for the joint embedding of entity and text. Ganea and Hofmann [123] 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 [124] developed an end-to-end neural architecture with relationwise and mentionwise normalization.

4) Entity Alignment: The tasks, as mentioned earlier, involve entity discovery from text or a single knowledge graph, while 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. 8(b).

Embedding-based alignment calculates the similarity between the embeddings of a pair of entities. MTransE [125]

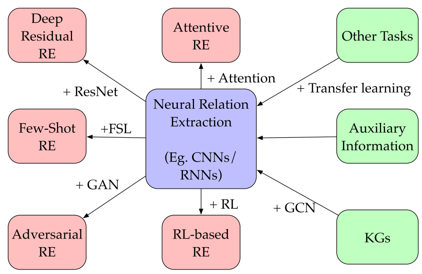


Fig. 9. Overview of NRE.

first studies EA in the multilingual scenario. It considers distance-based axis calibration, translation vectors, and linear transformations for cross-lingual entity matching and triple alignment verification. Following the translation-based and linear transformation models, IPTransE [126] proposes an iterative alignment model by mapping entities into a unified representation space under a joint embedding framework [see Fig. 8(b)] through aligned translation as $\|\mathbf{e}_1 + \mathbf{r}^{(\mathcal{E}_1 \to \mathcal{E}_2)} - \mathbf{e}_2\|$, linear transformation as $\|\mathbf{M}^{(\mathcal{E}_1 \to \mathcal{E}_2)}\mathbf{e}_1 - \mathbf{e}_2\|$, and parameter sharing as $\mathbf{e}_1 \equiv \mathbf{e}_2$. To solve error accumulation in iterative alignment, BootEA [127] 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 [128] capturing the correlation between cross-lingual attributes, KDCoE [129] embedding multilingual entity descriptions via cotraining, and MultiKE [130] learning multiple views of the entity name, relation, and attributes, and alignment with character attribute embedding [131]. EA has been intensively studied in recent years. We recommend Sun *et al.*'s quantitative survey [132] for detailed reading.

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 [133], 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.* [134] 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 [134], with a recent approach exploring the inner correlation between features [135]. Deep neural networks are changing the representation learning of knowledge graphs and texts. This section reviews recent advances in neural relation extraction (NRE), with an overview illustrated in Fig. 9.

1) Neural Relation Extraction: Trendy neural networks are widely applied to NRE. CNNs with position features of relative distances to entities [136] are first explored for relation classification and then extended to relation extraction by multiwindow CNN [137] with multiple sized convolutional filters. Multi-instance learning takes a bag of sentences as input to predict the relationship of the entity pair. PCNN [138] applies the piecewise max-pooling over the segments of convolutional representation divided by entity position. Compared with vanilla CNN [136], PCNN can more efficiently capture the structural information within the entity pair. MIMLCNN [139] further extends it to multilabel learning with cross-sentence max pooling for feature selection. Side information, such as class ties [140] and relation path [141], is also utilized. RNNs are also introduced; for example, SDP-LSTM [142] adopts multichannel LSTM while utilizing the shortest dependence path between entity pair, and Miwa and Bansal [143] stack sequential and tree-structure LSTMs based on dependence tree. BRCNN [144] combines RNN for capturing sequential dependence 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, including word-level attention to capture semantic information of words [145] and selective attention over multiple instances to alleviate the impact of noisy instances [146]. Other side information is also introduced for enriching semantic representation. APCNN [147] introduces entity description by PCNN and sentence-level attention, while HATT [148] 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 [80] proposes word-level attention with BiLSTM. Recently, Soares *et al.* [149] utilized pretrained relation representations from the deep transformer model.

3) Graph Convolutional Networks: GCNs are utilized for encoding a dependence tree over sentences or learning KGEs to leverage relational knowledge for sentence encoding. C-GCN [150] is a contextualized GCN model over the pruned dependence tree of sentences after path-centric pruning. AGGCN [151] also applies GCN over the dependence tree but utilizes multihead attention for edge selection in a soft weighting manner. Unlike the previous two GCN-based models, Zhang *et al.*, [152] 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 instances.

4) Adversarial Training: AT is applied to add adversarial noise to word embeddings for CNN- and RNN-based relation extractions under the MIML learning setting [153]. DSGAN [154] 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 NRE recently by training instance selectors with policy networks. Qin *et al.* [155] 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.* [156] and Feng *et al.* [157] 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 architecture for effective relation extraction. Recently, HRL [158] 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 NRE. Noticing that current NRE methods do not use very deep networks, Huang and Wang [159] applied deep residual learning to noisy relation extraction and found that nine-layer CNNs have improved performance. Liu *et al.* [160] proposed to initialize the neural model by transfer learning from entity classification. The cooperative CORD [161] ensembles text corpus and knowledge graph with external logical rules by bidirectional knowledge distillation and adaptive imitation. TK-MF [162] enriches sentence representation learning by matching sentences and topic words. Recently, Shahbazi *et al.* [163] studied trustworthy relation extraction by benchmarking several explanation mechanisms, including saliency, gradient $\times$ 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.* [164] proposed hybrid attention-based prototypical networks to compute prototypical relation embedding and compare its distance between the query embedding. Qin *et al.* [165] explored the relationships between relations with a global relation graph and formulated few-shot relation extraction as a Bayesian metalearning problem to learn the posterior distribution of relations' prototype vectors.

7) Joint Entity and Relation Extraction: Traditional relation extraction models utilize pipeline approaches by first extracting entity mentions and then classifying relations. However, pipeline methods may cause error accumulation. Several studies show better performance by joint learning [143], [166] than by conventional pipeline methods. Katiyar and Cardie [167] proposed a joint extraction framework with an attention-based LSTM network. Some convert joint extraction into different problems, such as sequence labeling via a novel tagging scheme [168] and multiturn question answering [169]. Challenges remain in dealing with entity pair and relation overlapping [170]. Wei *et al.* [171] proposed a cascade binary tagging framework that models relations as subject–object mapping functions to solve the overlapping problem.

There is a distribution discrepancy between training and inference in the joint learning framework, leading to exposure bias. Recently, Wang *et al.* [172] proposed a one-stage joint extraction framework by transforming joint entity and relation extraction into a token pair linking task to mitigate error propagation and exposure bias. In contrast to the common view that joint models can ease error accumulation by capturing mutual interaction of entities and relations, Zhong and Chen [173] proposed a simple pipeline-based yet effective approach to learning two independent encoders for entities and relations, revealing that strong contextual representation can preserve distinct features of entities and relations. Future research needs to rethink the relation between the pipeline and joint learning methods.

D. Summary

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

KGC 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 at the individual relation level and is poor at complex reasoning because it ignores the symbolical nature of the 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 metarelational 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 seeds. 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 [146] 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 III.

V. Temporal Knowledge Graph

Current knowledge graph research mostly focuses on static knowledge graphs where facts are not changed with time, while the temporal dynamics of a knowledge graph are 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 *temporal knowledge graph* in contrast to the previous static knowledge graph. Research efforts have been made for learning temporal and relational embeddings simultaneously. Relevant models for dynamic network embedding also inspire temporal KGE. For example, the temporal graph attention (TGAT) network [174] that captures temporal–topological structure and learn time-feature interactions simultaneously may be useful to preserve temporal-aware relation for knowledge graphs.

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 [175] investigated temporal scope prediction over time-annotated triple and simply extended existing embedding methods, for example, TransE with the vector-based TTransE defined as

$$f_\tau(h, r, t) = -\|\mathbf{h} + \mathbf{r} + \tau - \mathbf{t}\|_{L_{1/2}}. \tag{23}$$

Ma *et al.* [176] also generalized existing static embedding methods and proposed ConT by replacing the shared weight

TABLE III
SUMMARY OF NRE AND RECENT ADVANCES

Category	Method	Mechanism	Auxiliary Information
CNNs	O-CNN [136]	CNN + max pooling	position embedding
	Multi CNN [137]	Multi-window convolution + max pooling	position embedding
	PCNN [138]	CNN + piecewise max pooling	position embedding
	MIMLCNN [139]	CNN + piecewise and cross-sentence max pooling	position embedding
	Ye et al. [140]	CNN/PCNN + pairwise ranking	position embedding, class ties
	Zeng et al. [141]	CNN + max pooling	position embedding, relation path
RNNs	SDP-LSTM [142]	Multichannel LSTM + dropout	dependency tree, POS, GR, hypernyms
	LSTM-RNN [143]	Bi-LSTM + Bi-TreeLSTM	POS, dependency tree
	BRCNN [144]	Two-channel LSTM + CNN + max pooling	dependency tree, POS, NER
Attention	Attention-CNN [145]	CNN + word-level attention + max pooling	POS, position embedding
	Lin et al. [146]	CNN/PCNN + selective attention + max pooling	position embedding
	Att-BLSTM [80]	Bi-LSTM + word-level attention	position indicator
	APCNN [147]	PCNN + sentence-level attention	entity descriptions
	HATT [148]	CNN/PCNN + hierarchical attention	position embedding, relation hierarchy
GCNs	C-GCN [150]	LSTM + GCN + path-centric pruning	dependency tree
	KATT [152]	Pre-training + GCN + CNN + attention	position embedding, relation hierarchy
	AGGCN [151]	GCN + multi-head attention + dense layers	dependency tree
Adversarial	Wu et al. [153]	AT + PCNN/RNN + selective attention	indicator encoding
	DSGAN [154]	GAN + PCNN/CNN + attention	position embedding
RL	Qin et al. [155]	Policy gradient + CNN + performance change reward	position embedding
	Zeng et al. [156]	Policy gradient + CNN + +1/-1 bag-result reward	position embedding
	Feng et al. [157]	Policy gradient + CNN + predictive probability reward	position embedding
	HRL [158]	Hierarchical policy learning + Bi-LSTM + MLP	relation indicator

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 [177] takes a time stamp as a hyperplane $\mathbf{w}_\tau$ and projects entity and relation representation as $P_\tau(\mathbf{h}) = \mathbf{h} - (\mathbf{w}_\tau^\top \mathbf{h})\mathbf{w}_\tau$, $P_\tau(\mathbf{t}) = \mathbf{t} - (\mathbf{w}_\tau^\top \mathbf{t})\mathbf{w}_\tau$, and $P_\tau(\mathbf{r}) = \mathbf{r} - (\mathbf{w}_\tau^\top \mathbf{r})\mathbf{w}_\tau$. The temporally projected scoring function is calculated as

$$f_\tau(h, r, t) = \|P_\tau(\mathbf{h}) + P_\tau(\mathbf{r}) - P_\tau(\mathbf{t})\|_{L_1/L_2} \tag{24}$$

within the projected translation of $P_\tau(\mathbf{h}) + P_\tau(\mathbf{r}) \approx P_\tau(\mathbf{t})$. García-Durán *et al.* [178] 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 functions of extended TransE and DistMult are calculated as $\|\mathbf{h}+\mathbf{r}_{\text{temp}}-\mathbf{t}\|_2$ and $(\mathbf{h}\circ\mathbf{t})\mathbf{r}_{\text{temp}}^T$, respectively. By defining the context of an entity e as an aggregate set of facts containing e, Liu *et al.* [179] proposed context selection to capture useful contexts and measured temporal consistency with selected context. By formulating temporal KGC as four-order tensor completion, Lacroix *et al.* [180] proposed TComplEx that extends ComplEx decomposition and introduced weighted regularizers.

B. Entity Dynamics

Real-world events change entities' states and, consequently, affect the corresponding relations. To improve temporal scope inference, the contextual temporal profile model [181] 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.* [182] 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 [183], 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 nonlinear temporal evolution. To capture the interaction between nodes, RE-NET [184] models event sequences via an 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 Dependence

There exists temporal dependencies in relational chains following the timeline, for example, `wasBornIn` $\rightarrow$ `graduateFrom` $\rightarrow$ `workAt` $\rightarrow$ `diedIn`. Jiang *et al.* [185], [186] 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

$$f(\langle r_k, r_l\rangle) = \|\mathbf{r}_k\mathbf{T} - \mathbf{r}_l\|_{L_{1/2}} \tag{25}$$

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.* [187] explored Markov logic network and probabilistic soft logic for reasoning over uncertain temporal knowledge graphs. RLvLR-Stream [95] 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. The application of knowledge graphs includes twofold: 1) in-KG applications, such as link prediction and NER and 2) out-of-KG applications, including relation extraction and more downstream knowledge-aware applications, such as question answering and recommendation systems. This section introduces several recent DNN-based knowledge-driven approaches with applications on natural language processing and recommendation. More miscellaneous applications, such as digital health and search engine, are introduced in Appendix E in the Supplementary Material.

A. Language Representation Learning

Language representation learning via self-supervised language model pretraining has become an integral component of many NLP systems. 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. The knowledge graph language model (KGLM) [188] learns to render knowledge by selecting and copying entities. ERNIE-Tsinghua [189] fuses informative entities via aggregated pretraining and random masking. K-BERT [116] infuses domain knowledge into BERT contextual encoder. ERNIE-Baidu [190] introduces named entity masking and phrase masking to integrate knowledge into the language model and is further improved by ERNIE 2.0 [115] via continual multitask learning. To capture factual knowledge from text, KEPLER [191] combines knowledge embedding and masked language modeling losses via joint optimization. GLM [192] proposes a graph-guided entity masking scheme to utilize knowledge graph implicitly. CoLAKE [193] further exploits the knowledge context of an entity through a unified word-knowledge graph and a modified transformer encoder. Similar to the K-BERT model and focusing on the medical corpus, BERT-MK [194] integrates medical knowledge into the pretraining language model via knowledge subgraph extraction. Rethinking about large-scale training on language model and querying over knowledge graphs, Petroni *et al.* [195] analyzed the language model and knowledge base. They found that certain factual knowledge can be acquired via the pretraining 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. Dai *et al.* [196] proposed a conditional focused neural network equipped with focused pruning to reduce the search space. BAMnet [197] 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.* [198] found that sophisticated deep models, such as LSTM and GRU with heuristics, achieve state of the art, and nonneural models also gain reasonably well performance.

2) Multihop Reasoning: To deal with complex multihop relations, it requires a more dedicated design to be capable of multihop 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 multihop reasoning. Bauer *et al.* [199] proposed multihop bidirectional attention and pointer-generator decoder for effective multihop reasoning and coherent answer generation, utilizing external commonsense knowledge by relational path selection from ConceptNet and injection with selectively gated attention. The variational reasoning network (VRN) [200] conducts multihop logic reasoning with reasoning-graph embedding, while handling the uncertainty in topic entity recognition. KagNet [201] 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 [202] combines implicit extraction and explicit reasoning and proposes a cognitive graph model based on BERT and GNN for multihop QA.

C. Recommender Systems

Integrating knowledge graphs as external information enables recommendation systems to have the ability of commonsense reasoning, with the potential to solve the sparsity issue and the cold start problem. 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 [203] jointly trains KGEs, item's textual information, and visual content via translational KGE model and stacked autoencoders. Noticing that time- and topic-sensitive news articles consist of condensed entities and common knowledge, DKN [204] incorporates knowledge graph by a knowledge-aware CNN model with multichannel 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 [205] associates multitask 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 [206] 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 dependence. PGPR [207] performs reinforcement policy-guided path reasoning over knowledge-graph-based user–item interaction. KGAT [208] 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. Knowledge graph-based recommendation inherently processes interpretability from embedding propagation with multihop neighbors in the knowledge graph.

VII. Future Directions

Many efforts have been conducted to tackle the challenges of knowledge representation and its related applications. However, there remain 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 RL-based pathfinding and reasoning, are up-and-coming for handling complex reasoning. For the combination of logic rules and embeddings, recent works [102], [103] 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 [41] proved that HolE and ComplEx are mathematically equivalent for link prediction with a particular constraint. ANALOGY [22] provides a unified view of several representative models, including DistMult, ComplEx, and HolE. Wang *et al.* [47] explored connections among several bilinear models. Sharma *et al.* [209] 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.* [78] 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 toward unification in a way similar to the unified framework of graph networks [210], 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 [42] 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 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 tradeoff between computational efficiency and model expressiveness, with a limited number of works applied to more than one million entities. Several embedding methods use simplification to reduce the computation cost, such as simplifying tensor products with circular correlation operation [21]. 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 [102] are generated by simple brute-force search, making it insufficient on large-scale knowledge graphs. ExpressGNN [103] attempts to use NeuralLP [100] 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 pretraining via transformers and variants, such as BERT models. At the same time, a recent finding [195] reveals that the pretraining language model on the unstructured text can acquire certain factual knowledge. Large-scale pretraining 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 semiautomatic construction under the supervision of existing knowledge graphs. Facing 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 KRL, knowledge acquisition methods, and a wide variety of knowledge-aware applications. This article conducts a comprehensive survey on the following four scopes: 1) KGE, 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; and 4) real-world knowledge-aware applications on NLU, recommendation systems, question answering, and other miscellaneous applications. Besides, some useful resources of data sets 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 that it can facilitate future research.

References

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

[2] E. Shortliffe, *Computer-Based Medical Consultations: MYCIN*, vol. 2. Amsterdam, The Netherlands: Elsevier, 2012.

[3] X. Dong *et al.*, "Knowledge vault: A Web-scale approach to probabilistic knowledge fusion," in *Proc. SIGKDD*, 2014, pp. 601–610.

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

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

[6] A. Hogan *et al.*, "Knowledge graphs," 2020, *arXiv:2003.02320*. [Online]. Available: http://arxiv.org/abs/2003.02320

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

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

[9] Y. Lin, X. Han, R. Xie, Z. Liu, and M. Sun, "Knowledge representation learning: A quantitative review," 2018, *arXiv:1812.10901*. [Online]. Available: http://arxiv.org/abs/1812.10901

[10] R. H. Richens, "Preprogramming for mechanical translation," *Mech. Transl.*, 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, Dec. 2016.

[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, Sep. 2018.

[14] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112948.

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

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

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

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

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

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

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

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

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

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

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

[26] S. He, K. Liu, G. Ji, and J. Zhao, "Learning to represent knowledge graphs with Gaussian embedding," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 623–632.

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

[28] H. Xiao, M. Huang, Y. Hao, and X. Zhu, "From one point to a manifold: Orbit models for knowledge graph embedding," in *Proc. IJCAI*, 2016, pp. 1315–1321.

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

[30] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6901–6914.

[31] C. Xu and R. Li, "Relation embedding with dihedral group in knowledge graph," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 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 *Proc. 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 *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, 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 *Proc. 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 *Proc. 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 *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 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 *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 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 *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 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," *Mach. Learn.*, vol. 94, no. 2, pp. 233–259, Feb. 2014.

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

[41] K. Hayashi and M. Shimbo, "On the equivalence of holographic and complex embeddings for link prediction," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 554–559.

[42] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 96–104.

[43] 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 *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 2, 2018, pp. 327–333.

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

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

[46] Q. Wang *et al.*, "CoKE: Contextualized knowledge graph embedding," 2019, *arXiv:1911.02168*. [Online]. Available: http://arxiv.org/abs/1911.02168

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

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

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

[50] M. Nickel, V. Tresp, and H. P. Kriegel, "Factorizing YAGO: Scalable machine learning for linked data," in *Proc. WWW*, 2012, pp. 271–280.

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

[52] I. Balazevic, C. Allen, and T. Hospedales, "TuckER: Tensor factorization for knowledge graph completion," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 5185–5194.

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

[54] Q. Liu *et al.*, "Probabilistic reasoning via deep learning: Neural association models," 2016, *arXiv:1603.07704*. [Online]. Available: http://arxiv.org/abs/1603.07704

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

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

[57] M. Gardner, P. Talukdar, J. Krishnamurthy, and T. Mitchell, "Incorporating vector space similarity in random walk inference over knowledge bases," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 397–406.

[58] A. Neelakantan, B. Roth, and A. McCallum, "Compositional vector space models for knowledge base completion," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 156–166.

[59] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for knowledge graph completion," 2019, *arXiv:1909.03193*. [Online]. Available: http://arxiv.org/abs/1909.03193

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

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

[62] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4710–4723.

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

[64] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph and text jointly embedding," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1591–1601.

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

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

[67] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, "Semantically smooth knowledge graph embedding," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 84–94.

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

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

[70] Z. Zhang, F. Zhuang, M. Qu, F. Lin, and Q. He, "Knowledge graph embedding with hierarchical relation structure," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3198–3207.

[71] R. Xie, Z. Liu, H. Luan, and M. Sun, "Image-embodied knowledge representation learning," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3140–3146.

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

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

[74] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," in *Proc. AAAI*, 2017, pp. 4444–4451.

[75] X. Chen, M. Chen, W. Shi, Y. Sun, and C. Zaniolo, "Embedding uncertain knowledge graphs," in *Proc. AAAI*, vol. 33, 2019, pp. 3363–3370.

[76] P. Tabacof and L. Costabello, "Probability calibration for knowledge graph embedding models," in *Proc. ICLR*, 2019.

[77] T. Safavi, D. Koutra, and E. Meij, "Evaluating the calibration of knowledge graph embeddings for trustworthy link prediction," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 8308–8321.

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

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

[80] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2016, pp. 207–212.

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

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

[83] S. Guan, X. Jin, Y. Wang, and X. Cheng, "Shared embedding based neural networks for knowledge graph completion," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 247–256.

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

[85] C. Zhang, Y. Li, N. Du, W. Fan, and P. S. Yu, "On the generative discovery of structured medical knowledge," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2720–2728.

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

[87] R. Das, A. Neelakantan, D. Belanger, and A. McCallum, "Chains of reasoning over entities, relations, and text using recurrent neural networks," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 132–141.

[88] W. Chen, W. Xiong, X. Yan, and W. Y. Wang, "Variational knowledge graph reasoning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 1, 2018, pp. 1823–1832.

[89] W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A reinforcement learning method for knowledge graph reasoning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 564–573.

[90] R. Das *et al.*, "Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning," in *Proc. ICLR*, 2018, pp. 1–18.

[91] X. V. Lin, R. Socher, and C. Xiong, "Multi-hop knowledge graph reasoning with reward shaping," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3243–3253.

[92] 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 *Proc. NeurIPS*, 2018, pp. 6786–6797.

[93] C. Fu, T. Chen, M. Qu, W. Jin, and X. Ren, "Collaborative policy learning for open knowledge graph reasoning," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 2672–2681.

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

[95] P. G. Omran, K. Wang, and Z. Wang, "An embedding-based approach to rule learning in knowledge graphs," *IEEE TKDE*, vol. 33, no. 4, pp. 1348–1359, Apr. 2021.

[96] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 192–202.

[97] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Knowledge graph embedding with iterative guidance from soft rules," in *Proc. AAAI*, 2018, pp. 4816–4823.

[98] W. Zhang *et al.*, "Iteratively learning embeddings and rules for knowledge graph reasoning," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2366–2377.

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

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

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

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

[103] Y. Zhang *et al.*, "Efficient probabilistic logic reasoning with graph neural networks," in *Proc. ICLR*, 2020, pp. 1–20.

[104] W. Xiong, M. Yu, S. Chang, X. Guo, and W. Y. Wang, "One-shot relational learning for knowledge graphs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1980–1990.

[105] 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 *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3374–3379.

[106] M. Chen, W. Zhang, W. Zhang, Q. Chen, and H. Chen, "Meta relational learning for few-shot link prediction in knowledge graphs," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4217–4226.

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

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

[109] J. Baek, D. B. Lee, and S. J. Hwang, "Learning to extrapolate knowledge: Transductive few-shot out-of-graph link prediction," in *Proc. NeurIPS*, 2020, pp. 546–560.

[110] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 357–370, Dec. 2016.

[111] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2016, pp. 260–270.

[112] C. Xia *et al.*, "Multi-grained named entity recognition," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1430–1440.

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

[114] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, "A unified MRC framework for named entity recognition," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5849–5859.

[115] Y. Sun *et al.*, "ERNIE 2.0: A continual pre-training framework for language understanding," in *Proc. AAAI*, 2020, pp. 8968–8975.

[116] W. Liu *et al.*, "K-BERT: Enabling language representation with knowledge graph," in *Proc. AAAI*, 2020, pp. 1–8.

[117] 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 *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1825–1834.

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

[119] 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 *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1709–1719.

[120] Y. Zhao, A. Zhang, R. Xie, K. Liu, and X. Wang, "Connecting embeddings for knowledge graph entity typing," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6419–6428.

[121] H. Huang, L. Heck, and H. Ji, "Leveraging deep neural networks and knowledge graphs for entity disambiguation," 2015, *arXiv:1504.07678*. [Online]. Available: http://arxiv.org/abs/1504.07678

[122] W. Fang, J. Zhang, D. Wang, Z. Chen, and M. Li, "Entity disambiguation by knowledge and text jointly embedding," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 260–269.

[123] O.-E. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2619–2629.

[124] P. Le and I. Titov, "Improving entity linking by modeling latent relations between mentions," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 1595–1604.

[125] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1511–1517.

[126] H. Zhu, R. Xie, Z. Liu, and M. Sun, "Iterative entity alignment via joint knowledge embeddings," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4258–4264.

[127] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4396–4402.

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

[129] 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 *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3998–4004.

[130] Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, and Y. Qu, "Multi-view knowledge graph embedding for entity alignment," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 5429–5435.

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

[132] Z. Sun *et al.*, "A benchmarking study of embedding-based entity alignment for knowledge graphs," in *Proc. VLDB*, 2020.

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

[134] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. (AFNLP ACL-IJCNLP)*, vol. 2, 2009, pp. 1003–1011.

[135] J. Qu, D. Ouyang, W. Hua, Y. Ye, and X. Zhou, "Discovering correlations between sparse features in distant supervision for relation extraction," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 726–734.

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

[137] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proc. 1st Workshop Vector Space Model. Natural Lang. Process.*, 2015, pp. 39–48.

[138] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1753–1762.

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

[140] H. Ye, W. Chao, Z. Luo, and Z. Li, "Jointly extracting relations with class ties via effective deep ranking," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1810–1820.

[141] W. Zeng, Y. Lin, Z. Liu, and M. Sun, "Incorporating relation paths in neural relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1768–1777.

[142] 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 *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1785–1794.

[143] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 1105–1116.

[144] R. Cai, X. Zhang, and H. Wang, "Bidirectional recurrent convolutional neural network for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 756–765.

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

[146] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 2124–2133.

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

[148] X. Han, P. Yu, Z. Liu, M. Sun, and P. Li, "Hierarchical relation extraction with coarse-to-fine grained attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2236–2245.

[149] L. Baldini Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2895–2905.

[150] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2205–2215.

[151] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 241–251.

[152] N. Zhang *et al.*, "Long-tail relation extraction via knowledge graph embeddings and graph convolution networks," in *Proc. Conf. North*, 2019, pp. 3016–3025.

[153] Y. Wu, D. Bamman, and S. Russell, "Adversarial training for relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1778–1783.

[154] P. Qin, W. Xu, and W. Y. Wang, "DSGAN: Generative adversarial training for distant supervision relation extraction," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 496–505.

[155] P. Qin, W. Xu, and W. Y. Wang, "Robust distant supervision relation extraction via deep reinforcement learning," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 2137–2147.

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

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

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

[159] Y. Huang and W. Y. Wang, "Deep residual learning for weakly-supervised relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1803–1807.

[160] T. Liu, X. Zhang, W. Zhou, and W. Jia, "Neural relation extraction via inner-sentence noise reduction and transfer learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2195–2204.

[161] K. Lei *et al.*, "Cooperative denoising for distantly supervised relation extraction," in *Proc. COLING*, 2018, pp. 426–436.

[162] H. Jiang *et al.*, "Relation extraction using supervision from topic knowledge of relation labels," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 5024–5030.

[163] H. Shahbazi, X. Fern, R. Ghaeini, and P. Tadepalli, "Relation extraction with explanation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6488–6494.

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

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

[166] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1858–1869.

[167] A. Katiyar and C. Cardie, "Going out on a limb: Joint extraction of entity mentions and relations without dependency trees," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 917–928.

[168] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1227–1236.

[169] X. Li *et al.*, "Entity-relation extraction as multi-turn question answering," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1340–1350.

[170] D. Dai, X. Xiao, Y. Lyu, S. Dou, Q. She, and H. Wang, "Joint extraction of entities and overlapping relations using position-attentive sequence labeling," in *Proc. AAAI*, vol. 33, 2019, pp. 6300–6308.

[171] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1476–1488.

[172] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun, "TPLinker: Single-stage joint extraction of entities and relations through token pair linking," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 1572–1582.

[173] Z. Zhong and D. Chen, "A frustratingly easy approach for entity and relation extraction," 2020, *arXiv:2010.12812*. [Online]. Available: http://arxiv.org/abs/2010.12812

[174] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in *Proc. ICLR*, 2020, pp. 1–19.

[175] J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in *Proc. Companion The Web Conf. Web Conf. (WWW)*, 2018, pp. 1771–1776.

[176] Y. Ma, V. Tresp, and E. A. Daxberger, "Embedding models for episodic knowledge graphs," *J. Web Semantics*, vol. 59, Dec. 2019, Art. no. 100490.

[177] S. S. Dasgupta, S. N. Ray, and P. Talukdar, "Hyte: Hyperplane-based temporally aware knowledge graph embedding," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2001–2011.

[178] A. García-Durán, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4816–4821.

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

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

[181] D. T. Wijaya, N. Nakashole, and T. M. Mitchell, "CTPs: Contextual temporal profiles for time scoping facts using state change detection," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1930–1936.

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

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

[184] W. Jin, C. Zhang, P. Szekely, and X. Ren, "Recurrent event network for reasoning over temporal knowledge graphs," in *Proc. ICLR RLGM Workshop*, 2019, pp. 1–6.

[185] T. Jiang *et al.*, "Towards time-aware knowledge graph completion," in *Proc. COLING*, 2016, pp. 1715–1724.

[186] T. Jiang *et al.*, "Encoding temporal information for time-aware link prediction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2350–2354.

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

[188] 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 *Proc. ACL*, 2019, pp. 5962–5971.

[189] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1441–1451.

[190] Y. Sun *et al.*, "ERNIE: Enhanced representation through knowledge integration," 2019, *arXiv:1904.09223*. [Online]. Available: http://arxiv.org/abs/1904.09223

[191] X. Wang, T. Gao, Z. Zhu, Z. Liu, J. Li, and J. Tang, "KEPLER: A unified model for knowledge embedding and pre-trained language representation," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 176–194, Feb. 2020.

[192] T. Shen, Y. Mao, P. He, G. Long, A. Trischler, and W. Chen, "Exploiting structured knowledge in text via graph-guided representation learning," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 8980–8994.

[193] T. Sun *et al.*, "CoLAKE: Contextualized language and knowledge embedding," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 3660–3670.

[194] B. He *et al.*, "BERT-MK: Integrating graph contextualized knowledge into pre-trained language models," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2020, pp. 2281–2290.

[195] F. Petroni *et al.*, "Language models as knowledge bases?" in *Proc. EMNLP-IJCNLP*, 2019, pp. 2463–2473.

[196] Z. Dai, L. Li, and W. Xu, "CFO: Conditional focused neural question answering with large-scale knowledge bases," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 800–810.

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

[198] S. Mohammed, P. Shi, and J. Lin, "Strong baselines for simple question answering over knowledge graphs with and without neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, vol. 2, 2018, pp. 291–296.

[199] L. Bauer, Y. Wang, and M. Bansal, "Commonsense for generative multi-hop question answering tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4220–4230.

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

[201] B. Y. Lin, X. Chen, J. Chen, and X. Ren, "KagNet: Knowledge-aware graph networks for commonsense reasoning," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 2829–2839.

[202] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, "Cognitive graph for multi-hop reading comprehension at scale," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2694–2703.

[203] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 353–362.

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

[205] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2000–2010.

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

[207] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, and Y. Zhang, "Reinforcement knowledge graph reasoning for explainable recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 285–294.

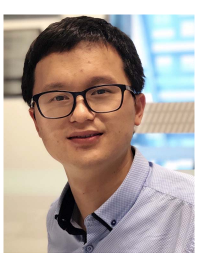
[208] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. SIGKDD*, 2019, pp. 950–958.

[209] A. Sharma and P. Talukdar, "Towards understanding the geometry of knowledge graph embeddings," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 122–131.

[210] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*. [Online]. Available: http://arxiv.org/abs/1806.01261

Shaoxiong Ji received the bachelor's degree in electronics and information engineering from the School of Information and Communication Engineering, Dalian University of Technology, Dalian, China, in 2014. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Aalto University, Aalto, Finland.

He worked as a Research Assistant at the University of Technology Sydney (UTS), Ultimo, NSW, Australia, The University of Queensland (UQ), Brisbane, QLD, Australia, and Nanyang Technological University (NTU), Singapore. His research interests include machine learning, data mining, and knowledge representation and reasoning.



Shirui Pan (Member, IEEE) received the Ph.D. degree in computer science from the University of Technology Sydney (UTS), Ultimo, NSW, Australia.

He was a Lecturer with the School of Software, UTS. He is currently a Lecturer with the Faculty of Information Technology, Monash University, Clayton, VIC, Australia. To date, he has published over 80 research articles in top-tier journals and conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), and the Annual Conference on Neural Information Processing Systems (NeurIPS). His research interests include data mining and machine learning.

Erik Cambria (Fellow, IEEE) received the Ph.D. degree through a joint program between the University of Stirling, Stirling, U.K., and the Media Lab, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2012.

He worked at Microsoft Research Asia, Beijing, China, and HP Labs India, Bengaluru, India. He is the Founder of SenticNet, a Singapore-based company offering B2B sentiment analysis services, and an Associate Professor with Nanyang Technological University (NTU), Singapore, where he also holds the appointment of the Provost Chair in Computer Science and Engineering.

Dr. Cambria is involved in many international conferences as a PC member and the Program Chair. He is also an Associate Editor of several journals, e.g., *Neurocomputing* (NEUCOM), *Information Fusion* (INFFUS), *Knowledge-Based Systems* (KBS), *IEEE Computational Intelligence Magazine* (IEEE CIM), and *IEEE Intelligent Systems* (where he manages the Department of Affective Computing and Sentiment Analysis).

Pekka Marttinen is currently an Assistant Professor of machine learning with Aalto University, Aalto, Finland, where he is also a Principal Investigator with the Machine Learning for Health (Aalto-ML4H) Research Group. His research interests include probabilistic machine learning, deep learning, Bayesian modeling, and applications in genomics and healthcare. He has coauthored 49 articles on these topics.

Philip S. Yu (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA.

He is currently a Distinguished Professor of computer science with the University of Illinois at Chicago, Chicago, IL, USA, where he is the Wexler Chair in Information Technology. He has published more than 830 articles in refereed journals and conferences. He holds or has applied for more than 300 U.S. patents. His research interests include big data, data mining, data streams, databases, and privacy.

Dr. Yu is a fellow of the ACM. He received the ACM SIGKDD 2016 Innovation Award, the Research Contributions Award from the IEEE International Conference on Data Mining in 2003, and the Technical Achievement Award from the IEEE Computer Society in 2013.