

Rule Learning over Knowledge Graphs: A Review

Hong Wu  

School of Information and Communication Technology, Griffith University, Australia

Zhe Wang  

School of Information and Communication Technology, Griffith University, Australia

Kewen Wang¹  

School of Information and Communication Technology, Griffith University, Australia

Pouya Ghiasnezhad Omran  

School of Computing, The Australian National University, Australia

Jiangmeng Li  

Institute of Software, Chinese Academy of Sciences, China

Abstract

Compared to black-box neural networks, logic rules express explicit knowledge, can provide human-understandable explanations for reasoning processes, and have found their wide application in knowledge graphs and other downstream tasks. As extracting rules manually from large knowledge graphs is labour-intensive and often infeasible, automated rule learning has recently attracted significant interest, and a number of approaches to rule learning for knowledge graphs have been proposed. This survey aims to provide a review of approaches and a classification of state-of-the-art

systems for learning first-order logic rules over knowledge graphs. A comparative analysis of various approaches to rule learning is conducted based on rule language biases, underlying methods, and evaluation metrics. The approaches we consider include inductive logic programming (ILP)-based, statistical path generalisation, and neuro-symbolic methods. Moreover, we highlight important and promising application scenarios of rule learning, such as rule-based knowledge graph completion, fact checking, and applications in other research areas.

2012 ACM Subject Classification Computing methodologies → Knowledge representation and reasoning; Information systems → Data mining

Keywords and phrases Rule learning, Knowledge graphs, Link prediction

Digital Object Identifier 10.4230/TGDK.1.1.7

Category Survey

Acknowledgements The authors would like to thank the editors and the anonymous referees for their constructive comments that have helped improve the quality of this paper.

Received 2023-07-02 **Accepted** 2023-08-31 **Published** 2023-12-19

Editors Aidan Hogan, Ian Horrocks, Andreas Hotho, and Lalana Kagal

Special Issue Trends in Graph Data and Knowledge

1 Introduction

Knowledge graphs are a popular form of knowledge bases that describe facts about real-world entities and their relations. They serve as powerful tools for organising and modelling information in a way that allows for efficient storage, retrieval, and reasoning. Recently, they have garnered significant attention in both academia and industry. Many public knowledge graphs have been developed, such as Freebase [8], WordNet [46], YAGO [69], DBPedia [3], and WikiData [76]. They

¹ The corresponding author.

highlight the pivotal role these graphs play as expansive and valuable resources, supporting a wide array of applications in artificial intelligence, data analysis, and knowledge representation. Additionally, numerous commercial knowledge graphs have been created, such as Google KG [18], Microsoft Satori [34], and Facebook Graph Search [19]. These knowledge graphs have demonstrated their capability to provide more efficient services for other products in their companies. They enable efficient querying and reasoning, allowing users and applications to gain valuable insights and make informed decisions based on interconnected knowledge. Many knowledge graphs are large-scale with millions of entities and facts. For instance, in the case of the DBpedia Core Release, the 2016-04 edition of the DBpedia dataset, it contains 6.0 million entities and 9.5 billion triples. These numbers underscore the substantial growth in both the size and complexity of knowledge graphs. Some medium-sized knowledge graphs encompass tens of thousands of entities and several hundreds of thousands of facts.

The rise of knowledge graphs is intricately linked to the advancement of the Semantic Web [7]. Abiding by the triple-based definition of the Semantic Web, a knowledge graph is a set of RDF triples such as $(\text{Allen}, \text{livesIn}, \text{NewYork})$, which means Allen lives in the city of New York. As highlighted by some researchers [49, 6], a knowledge graph goes beyond being a simple graph database. Data management and reasoning in knowledge graphs can be empowered by a layer of conceptual knowledge, known as the ontology layer, and such conceptual knowledge can be represented as logic rules. For instance, consider a rule $0.9 : \text{hasBirthPlace}(x, y) \wedge \text{isRegionOf}(y, z) \rightarrow \text{hasNationality}(x, z)$. This rule implies that if x was born in the place y of country z , then x may have the nationality of z with a confidence degree of 0.9. Such rules can be used for reasoning over knowledge graphs, enabling the prediction of new facts based on existing ones. Unlike black-box deep neural networks, rules offer explicit high-level knowledge and can provide human-understandable explanations for the reasoning processes. However, crafting rules manually for large knowledge graphs is challenging, if not practically impossible. Therefore, the automatic extraction of high-level rules becomes both useful and important for knowledge graphs. The learned rules can be directly applied to reasoning in knowledge graph completion tasks. Additionally, they can be indirectly utilised to enhance the interpretability of neural network models for knowledge graph reasoning.

Learning Horn clauses has been studied extensively in the *inductive logic programming (ILP)* literature [47, 47, 15]. In the context of ILP-based approaches, first-order Horn rules are explored by systematically exploring the rule space through various refinement operators. Classical ILP systems [92, 68] cannot be used directly to handle knowledge graphs due to the lack of negative examples and the large data sizes. Recently, ILP-based rule learners such as AMIE [24] and its extensions [23, 37] have been developed with the aim of handling knowledge graphs. Thus, predicates in rules learned by AMIE+ are binary. This language bias helps significantly reduce the search space in rule learning. They use plausibility metrics adapted from association rule mining to address the lack of negative examples. ILP-based rule learners for knowledge graphs usually assign a confidence score for each learned rule. Such approaches are also referred to as *Probabilistic Inductive Logic Programming (PILP)* in the literature [65]. Another group of approaches generate candidate rules by directly exploring frequent patterns or paths of different granularity. A typical system in this group is AnyBURL [45, 44], which samples paths within knowledge graphs and generalises them to form rules. It generalises path instances by substituting entities from sampled paths with variables, thus forming rule patterns. The effectiveness of such an approach relies on the ability to sample representative paths and apply suitable statistical metrics. Apart from instance-level paths, some other rule learners explore ontological-level paths within knowledge graphs [13, 12, 56], which significantly reduces the search space of paths. Recently, there has been an emerging interest in integrating neural networks into the realm of rule learning. Neural

networks have shown remarkable success in various machine learning tasks and can automatically learn feature representations from raw data, including knowledge graphs. These neuro-symbolic approaches have the advantage of simultaneously learning both rule structures and parameters. Neural LP [89] is the first attempt to propose a framework combining both of the learning in an end-to-end differentiable model. Another group combines neural network models with other rule-learning strategies through knowledge graph embeddings, like EMBEDRULE [88] and RLvLR [51]. The incorporation of embedding can improve the scalability of rule learning over large knowledge graphs.

In this paper, we survey major approaches to learning first-order Horn rules over knowledge graphs, aiming to serve as a resource for researchers and practitioners in rule discovery over knowledge graphs. We are unaware of any similar survey paper on automated rule learning over knowledge graphs. More specifically, this paper provides a comprehensive review of state-of-the-art rule learners. It presents a comparative analysis of various approaches to rule learning, considering factors such as language bias, evaluation metrics, and underlying methods. Secondly, a categorisation of rule learning methods and techniques is provided. The surveyed approaches encompass three main categories: ILP-based, statistical path generalisation, and neuro-symbolic methods. Thirdly, this survey investigates the important and promising application scenario of logic rules, offering valuable insights into the current and future directions of this important field. By providing a comprehensive overview of the state-of-the-art approaches and highlighting the challenges and opportunities in rule learning, we hope to inspire further research and innovations in this area. We believe that this survey will facilitate knowledge exchange and collaboration among scholars and industry professionals, ultimately leading to significant contributions to the field of rule learning and beyond.

The rest of this paper is organised as follows. Section 2 provides an overview of our survey paper, introducing the definition of knowledge graphs, first-order Horn rules, a classification of learning methods, and the rule evaluation metrics. Sections 3, 4 and 5 focus on three categories of rule learning methods, respectively. Section 6 introduces the applications of first-order Horn rules automatically learned by rule learners. Finally, we discuss the future directions of rule learning and conclude the paper in Section 7.

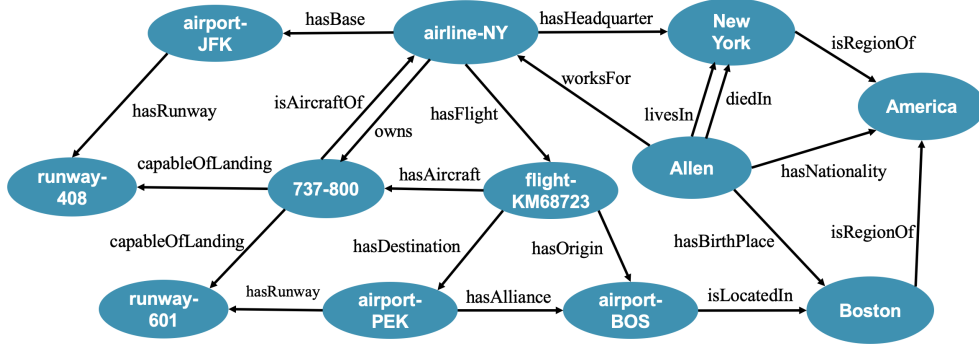
2 Overview

In this section, we will first fix some definitions and notations in knowledge graphs and rule languages that will be used in the paper. Then, we formulate the problem of rule learning in knowledge graphs, and then propose a classification of rule learning methods. A method of rule learning is essentially a process of ranking candidate rules. So, we will also introduce three confidence measures of rules before we discuss specific methods of rule learning later.

2.1 Knowledge Graphs and Rules

In this subsection, we introduce the basics of knowledge graphs and rules.

Knowledge graphs represent real-world entities, such as persons and places, and binary relations among them. A *knowledge graph* (KG) is often expressed as a set of *triples* of the form (s, p, o) , where entities s and o are called the subject and object of the triple, respectively, and p is the relation. A KG is essentially a directed multi-relational graph by viewing the entity (the subjects and objects) as the vertices and a triple (s, p, o) as an edge from s to o with the label p . For instance, a triple (airline-NY, hasBase, airport-JFK) describes that the two entities airline-NY and airport-JFK are connected by the relation hasBase. Following the convention in knowledge representation, a triple is also denoted as a fact $p(s, o)$.



■ **Figure 1** An example knowledge graph.

Formally, let \mathcal{E} and \mathcal{P} be respectively the sets of entities and relations in a KG \mathcal{G} . For a relation p , p^{-} denotes its *inverse*, i.e., triple $(\text{runway-408}, \text{hasRunway}^{-1}, \text{airport-JFK})$ is equivalent to $(\text{airport-JFK}, \text{hasRunway}, \text{runway-408})$. And \mathcal{P}^* denotes the set of the relations and their inverse relations, i.e., $\mathcal{P} \cup \{p^{-} \mid p \in \mathcal{P}\}$. In this paper, we consider the class of *first-order Horn rules*, which is sufficiently expressive for many practical applications in the Semantic Web and AI, and allows efficient reasoning algorithms. Moreover, in KGs, only binary and unary relations are considered. While a binary relation connects two entities, a unary relation represents a type (or class) of entities. In first-order logic, a relation is expressed as a predicate. Whenever no confusion is caused, we use these two terms alternatively.

A term is either an entity or a variable. If p is a binary relation, $p(t_1, t_2)$ is an atom, where t_1 and t_2 are terms. Similarly, if p is a unary relation, $p(t)$ is an atom, where t is a term.

A *first-order Horn rule* r is of the form

$$b_1 \wedge \dots \wedge b_n \rightarrow h \quad (1)$$

where h, b_1, \dots, b_n are atoms. The atom h is the head of r , denoted $head(r)$, and the conjunction of atoms b_1, \dots, b_n is the body of r , denoted $body(r)$. Intuitively, the rule r reads that if b_1, \dots, b_n hold, then h holds too. The *length* of the above rule body is n (the number of body atoms in the rule).

Due to the enormous search space of first-order Horn rules over large KGs, existing rule learning approaches often adopt certain language biases to restrict the forms of rules to learn, such as constraining the maximum length of rules, to effectively reduce the search space. This enables the rule-learning algorithm to be more efficient and applicable in practical scenarios. These constraints strike a balance between the size of the search space and the expressiveness of rules.

A most common language bias is to learn rules that represent path patterns in KGs, that is, the class of *closed-path (CP)* rules [81, 51]. Intuitively, in a CP rule, the body atoms form a path from the subject to the object of the head atom (involving only variables not entities). Formally, a *closed-path* rule is of the form

$$p_1(x_0, x_1) \wedge p_2(x_1, x_2) \wedge \dots \wedge p_n(x_{n-1}, x_n) \rightarrow p(x_0, x_n), \quad (2)$$

where $p \in \mathcal{P}$, $p_i \in \mathcal{P}^*$ ($1 \leq i \leq n$) and x_j 's ($0 \leq j \leq n$) are variables. Note that CP rules allow recursion, i.e., the head predicate can occur in the body. The advantage of closed-path rules lies in their ability to capture specific and meaningful patterns in the data. These rules can reveal intricate dependencies, cyclic patterns, and sequential behaviours present in the data, providing

deeper insights into the underlying associations. Focusing on CP rules reduces the search for candidate rules to the problem of path finding and ranking in KGs. This type of rules have been widely adopted by major rule learners [88, 25, 13, 51, 56].

As the class of CP rules is too limited for some applications, one way to expand the class of CP rules is to allow rules that are *closed* and *connected*. Two atoms in a rule are connected if they share a variable or an entity. A rule is connected if every atom in the rule is connected to another atom. A variable in a rule is closed if it appears at least twice in the rule. A rule is closed if all of its variables are closed.

To avoid learning rules with unrelated atoms, some methods, such as AMIE and its variants [24, 23, 37], require that the graphs of learned rules to be *connected*. In addition, to avoid having variables with existential quantifier in the rule head, a learned rule must be *closed*. For example, the rule $p_1(x_0, x_1) \wedge p_2(x_1, x_2) \rightarrow p(x_0, x_1)$ is connected but not closed, while the rule $p_1(x_0, x_1) \wedge p_2(x_2, x_2) \rightarrow p(x_0, x_1)$ is closed but not connected. By the definition, it is clear that a CP rule is both connected and closed, but not vice versa. For example, $p_1(x_0, x_1) \wedge p_2(x_1, x_2) \wedge p_3(x_0, x_1) \rightarrow p(x_0, x_2)$ is both connected and closed, but not a CP rule.

Another natural extension of CP rules is to allow unary predicates, i.e., classes (or types), in the rules [82]. Such a rule also describes a path pattern in the KG, but allows to specify the classes (or types) of the nodes on the paths. The rule $\text{person}(x_0) \wedge \text{hasBirthPlace}(x_0, x_1) \wedge \text{city}(x_1) \wedge \text{isRegionOf}(x_1, x_2) \wedge \text{country}(x_2) \rightarrow \text{hasNationality}(x_0, x_2)$ is a typed rule. It specifies the classes (types) of x_0, x_1, x_2 to be Person, City, Country.

Some rule learners can learn more expressive rules beyond first-order Horn rules, allowing negations [21, 31, 75], numeric values [54, 77], temporal values [52, 42], etc. Such extensions are useful for practical applications. Specifically, some approaches can learn rules that involve comparisons among numeric values, for example, $\text{hasBirthYear}(x, v_0) \wedge \text{hasBirthYear}(y, v_1) \wedge v_0 > v_1 \rightarrow \text{younger}(x, y)$. This rule says if a person x was born after another person y , then x is younger than y . Here, the relation hasBirthYear takes literal numbers as its object datatype. Some approaches focus on learning nonmonotonic rules (or negated rules) with negated atoms in the rule body, such as $\text{bornIn}(x, y) \wedge \text{not immigrate}(x, z) \rightarrow \text{livesIn}(x, y)$ says that a person x who was born in a place y and is not known to have migrated to z lives in y . Moreover, temporal rules can be learned over temporal KGs where every atom has a timestamp. For example, the rule $\text{bornIn}(x, y, t) \rightarrow \text{diedIn}(x, y, t + 80)$ indicates that if a person x born in city y at timestamp t usually die in the same place at time $t + 80$.

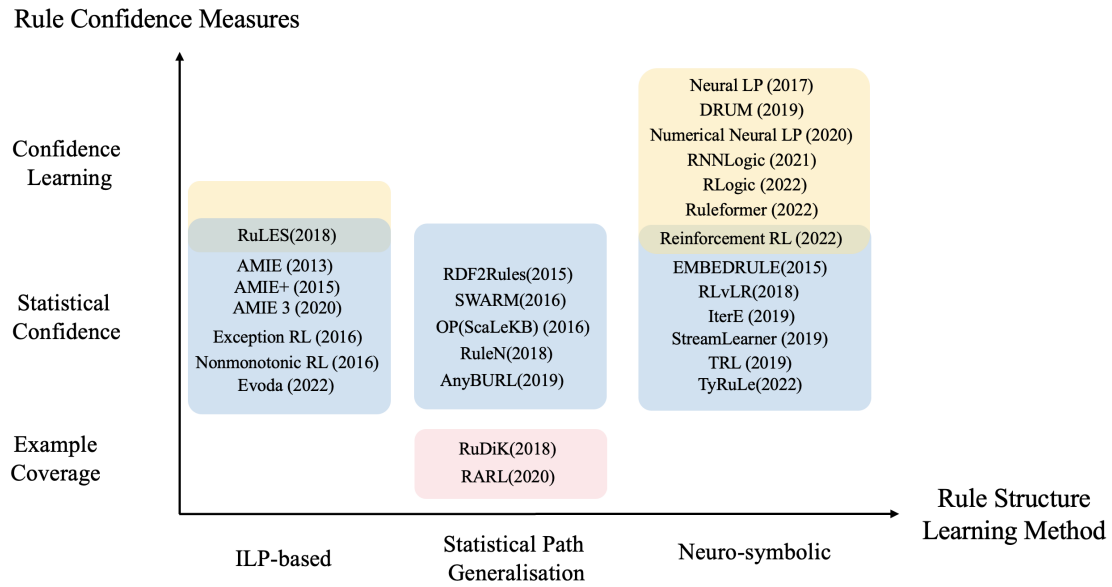
2.2 A Classification of Rule Learning Methods

The task of *rule learning* is to automatically extract a set of first-order logic Horn rules over a given KG. Formally, given a KG \mathcal{G} , a rule learning system, a.k.a. *rule learner*, learns a set of rules r of the form $b_1 \wedge \dots \wedge b_n \rightarrow h$ with a confidence degree $0 \leq \alpha_r \leq 1$ associated with each rule r . The relations and (possibly) entities in r are from \mathcal{G} , and r is considered plausible if there are many instances of r obtained by substituting the variables in r with entities in \mathcal{G} , such that the atoms in these instances are facts occurring in the KG \mathcal{G} . The more such instances exist the more plausible r is. The confidence degree α_r is calculated to reflect the plausibility of r , i.e., the more plausible r is the higher α_r should be.

Rule learning involves both learning the rule structures and estimating their plausibility. For example, in traditional Inductive Logic Programming, rule structure learning is achieved by systematically exploring the rule space by adding, updating, or deleting atoms at a time in the rule bodies. Rule plausibility is measured via example coverage by the set of learned rules. That is, given two sets of positive examples (true facts in the data) and negative examples (false or absent facts), the rule learner aims to induce a set of rules that cover as many positive examples

and as few negative examples as possible. Yet, due to the large sizes and the lack of negative examples in KGs, the traditional ILP methods of traversing rule space for structure learning and measuring example coverage for plausibility estimation are not directly applicable to KGs.

To handle the large sizes, complexity, incompleteness and dynamics of KGs, many novel and efficient rule learners have been developed. We classify them by two dimensions as shown in Figure 2, according to their structure learning and confidence measure methods.



■ **Figure 2** The categorisation of rule learners.

According to the rule structure learning methods, we can broadly categorise existing methods into three groups: the *inductive logic programming (ILP)-based* approaches, which use refinement operators to guide the search in the rule space, the *statistical path generalisation* methods, which extract frequent patterns from sampled paths or sub-graphs of the KGs, and the *neuro-symbolic* approaches, which directly or indirectly utilise neural networks to learn rules. We will introduce these three groups of methods with their representative works in Sections 3, 4, and 5.

According to the confidence measures, we classify the existing methods into three groups: the *example coverage* measures, which are in the same spirit as ILP approaches, by generating negative examples from KGs via a form of closed world assumption; the *statistical confidence* measures, which adapt statistical measures from association rule mining to address the lack of negative examples; and the *confidence learning* approaches, which learn confidence degrees as parameters of some neural networks. We will discuss these confidence measures in further detail in the remainder of this section.

Generally speaking, earlier works on KG rule learning are mostly ILP-based or statistical path generalisation methods, while most recent ones are largely neuro-symbolic methods. Statistical confidence measures are the most widely used, as they do not rely on the existence of negative examples and have better interpretability, i.e., statistical meanings. While confidence learning is only adopted by neuro-symbolic methods, example coverage measures have also been employed in statistical path generalisation methods. Interestingly, we have not found an ILP-based rule learner for KGs that adopts the example coverage measure which originates from ILP.

2.3 Rule Confidence Measures

Rule confidence measures play a crucial role in rule learning, as they indicate the plausibility of the learned rules, and the accuracy of such measures is critical for downstream tasks based on the learned rules. There are three main approaches for rule confidence measures, example coverage, statistical, and confidence learning measures.

2.3.1 Example Coverage

Traditional rule learning approaches, such as inductive logic programming (ILP), typically measure the confidence of rules via their coverage of observed positive and negative examples. It views a collection of rules as a classification model for the observed examples, and the goal of rule learning is to find a set of rules that cover as many positive examples as possible and as few or zero negative examples as possible. Specifically, for a first-order Horn rule r , intuitively a *positive example* is a tuple of entities \mathbf{a} that satisfy the head of the rule $head(r)$, i.e., when the variables \mathbf{x} in h are substituted with \mathbf{a} , the resulting atom is a fact in the KG. Similarly, a *negative example* is a tuple of entities \mathbf{a}' that do not satisfy the head of the rule $head(r)$, i.e., the substituted atom is not a fact in the KG. The body of r *covers* (or simply say r *covers*) a tuple of entities \mathbf{a} if when \mathbf{x} is substituted with \mathbf{a} , there is a way to substitute the other variables in r to make all the facts obtained from the body of r facts in the KG.

Given two sets \mathcal{A}^+ and \mathcal{A}^- of respectively positive and negative examples, let $ex_{\mathcal{A}^+}(r)$ and $ex_{\mathcal{A}^-}(r)$ consist of respectively positive and negative examples covered by r . Formally, for a set of rules \mathcal{R} with the same head h , the *coverage* of \mathcal{R} w.r.t. \mathcal{A}^+ and \mathcal{A}^- is defined as

$$cover(\mathcal{R}, \mathcal{A}^+, \mathcal{A}^-) = \beta \cdot \frac{|\bigcup_{r \in \mathcal{R}} ex_{\mathcal{A}^+}(r)|}{|\mathcal{A}^+|} - (1 - \beta) \cdot \frac{|\bigcup_{r \in \mathcal{R}} ex_{\mathcal{A}^-}(r)|}{|\mathcal{A}^-|}. \quad (3)$$

Some rule learning approaches measure the confidence of individual rules based on variants of the coverage [54, 56], by applying it to individual rules r instead of a rule set \mathcal{R} . However, a major challenge is that KGs adopt the Open World Assumption (OWA), that is, missing triples are not necessarily false but just unknown, which makes them inherently lack negative examples. To address this issue, certain negative examples generation strategies are employed [54, 56]. For example, negative examples are generated in [54] by considering (s, p, o) a negative example if it does not occur in the KG and at least one triple of the form (s, p, o') , (s, p', o) or (s', p, o) occurs in the KG, essentially following the PCA assumption (will be discussed in Section 2.3.2). Although several safeguards are employed in the negative example generation processes, they may still introduce noisy examples.

2.3.2 Statistical Confidence

Inspired by association rule mining and viewing logic rules as frequent patterns in KGs, some statistical measures such as *support* and *confidence* have been adapted for rule learning over KGs [24, 23, 37]. One advantage of such measures is that negative examples are not required.

For a first-order Horn rule r of the form (1), $ex_H(r)$ consists of all the tuples of entities \mathbf{a} that satisfy the head of the rule h . Similarly, $ex_B(r)$ consists of all the tuples of entities \mathbf{a} covered by the body of r , i.e., when the variables \mathbf{x} in the head h is substituted with \mathbf{a} , there is a way to substitute the other variables in r to make all the facts obtained from the body of r facts in the KG. Then, the *support* of r is defined as $supp(r) = |ex_H(r) \cap ex_B(r)|$. That is, the support of r is defined as the number of entities that satisfy both the head and the body of r . The *standard confidence* (SC) and *head coverage* (HC) of r are defined as follows

$$sc(r) = \frac{|ex_H(r) \cap ex_B(r)|}{|ex_B(r)|} \quad \text{and} \quad hc(r) = \frac{|ex_H(r) \cap ex_B(r)|}{|ex_H(r)|} \quad (4)$$

SC is the normalisation of support through the number of entity pairs that satisfy the body, while HC is the normalisation of support through the number of entity pairs that satisfy the head. The higher the values are for these measures, the more plausible the rule is.

These statistical measures have been widely applied to estimate the plausibility of rules learned from KGs [23, 13, 51, 31, 44, 94, 71]. And several variants of these measures have been proposed for better estimations.

It is argued that the standard confidence (SC) is not suitable for KGs that are highly incomplete. Some approaches refine it by introducing *partial completeness assumption (PCA)* [24, 23, 37]. PCA assumes the KG has complete information about an entity w.r.t. a relation if the KG contains at least one fact about the entity and relation. For example, if it is mentioned in KG that (Allen, worksFor, airline-NY), then we assume that he only has one job in New York airline. Some other variants along this line include *completeness confidence* [72, 73], which proposes to rely on explicitly incompleteness information to determine an instance as a counterexample, and *soft confidence* [81] which refines the SC with the entity type information. These statistical measures have also been extended to other forms of rules, such as negated rules [21, 75, 31], numerical rules [54, 77], temporal rules [52], and typed rules [82].

2.3.3 Confidence Learning

Traditionally, rule learning can be divided into two main steps: rule structure learning and rule confidence estimation. Recently, some approaches based on neural networks have been developed to perform rule structure and confidence learning simultaneously [89, 77, 62, 14, 87]. These approaches learn the confidence degree of a rule r as a learnable parameter $\alpha_r \in [0, 1]$, which will be updated during the training of the neural networks. This allows the trained neural networks to score any potential rules in the language bias and tightly couples the rule structure learning and confidence learning.

These approaches typically model the learning process through rule-based reasoning tasks, e.g., using rules to infer a triple. Hence, the objective function of the neural network is to assess the plausibility of each possible triple so that those existing in the KG have the highest plausibility. The input of the neural networks includes some latent representations (called embeddings) of the entities and relations in the KG, denoted \mathbf{e} for $e \in \mathcal{E}$ and \mathbf{p} for $p \in \mathcal{P}$. For a triple (s, p, o) and a rule r with its confidence α_r , let $score(\mathbf{s}, \mathbf{p}, \mathbf{o}, r)$ be a scoring function defined on the embeddings and the rule to score the plausibility of the triple. The neural networks are trained to update the parameters α_r that maximize scores for all the triples in the KG \mathcal{G} . This allows the neural networks to simultaneously learn both rule structure r and the rule confidence degrees α_r . By learning the confidence of rules as a parameter, the model can capture the uncertainty and ambiguity in the data, allowing it to assign appropriate confidence to different rules. This adaptability makes the rule-learning process more data-driven and helps the model to make more accurate predictions and better generalisations on unseen data. We will defer the detailed discussions of neural-based structure and parameter learning to Section 5.

In the following sections, we will talk about the three major groups of rule learning methods: LLP-based, statistical path generalisation, and neuro-symbolic methods.

3 ILP-based Methods

The classical ILP methods learn rule structures by systematically exploring the rule space (either in a top-down or a bottom-up manner) through refinement operators, such as adding, updating, or deleting one atom at a time in the rule bodies. However, existing ILP rule learners face at least two challenges when they are applied to KGs. First, the traditional rule search methods have difficulties in scaling to large KGs due to their high computational complexity. A KG can be much larger in size than datasets typically considered in the ILP literature. As a result, classical ILP systems like ALEPH [68] and QuickFOIL [92] cannot be used directly to perform rule learning over medium-sized or large KGs. Moreover, in traditional ILP settings, both positive and negative examples are normally provided for evaluating the rule coverage. However, KGs only contain positive facts, and obtaining negative examples becomes more challenging due to the *Open World Assumption* (OWA), which assumes that facts missing from the KG are not necessarily false but just unknown.

In recent years, many ILP-based rule learning methods have been proposed in response to these challenges. The AMIE series are notable representatives in this line of research, which introduces novel rule confidence measures to avoid the step of generating negative examples. Another line of research within ILP addresses the lack of negative examples by generating them. They focus on learning rules that can express exceptions or negations, known as nonmonotonic logic programs.

3.1 Without Negative Examples

As traditional ILP methods cannot handle rule learning due to the scale of search space and the lack of negative examples, recent ILP-based rule learners such as AMIE [24, 23, 37] tackle these challenges in rule learning over KGs by employing certain language biases and new confidence measures adapted from association rule mining. AMIE, especially its extension AMIE+ [23] is one of the earliest and most widely referenced KG rule learners.

To reduce the search space, AMIE learns *connected* and *closed* rules. The rule structure learning process of AMIE is a standard ILP top-down search to explore the rule search space. Top-down rule search starts with a general rule and then refines it by progressively adding more atoms. Based on defined refinement operators, such as adding atoms to make the rule closed and connected, it iteratively expands rules. Instead of using standard confidence (SC), AMIE calculates rule confidence under partial completeness assumption (PCA). If the rule satisfies the confidence measure thresholds, the rule is selected as a candidate rule.

AMIE+ and AMIE 3 [37] are the extensions of AMIE with a series of improvements and optimisations that allow the system to run over large-scale KGs. Specifically, AMIE+ speeds up the rule refinement phase for specific kinds of rules, simplifies the query of support, and approximates the PCA computations by an upper bound; AMIE 3 utilises an in-memory database and parallel computation to store and process large-scale KGs.

Some other rule learning methods use enhanced refinement operators for rule search. For example, Evoda [84] uses a *Genetic Logic Programming* algorithm that is combined with *Evolutionary Algorithms* (EA) to define refinement operators. Hence, three rule transformation operators are proposed, *mutation*, *crossover* and *selection*. This allows Evoda to learn rules that are not necessarily closed, e.g., it can learn a rule like $\text{owns}(x_0, x_1) \wedge \text{capableOfLanding}(x_1, x_2) \rightarrow \text{isAircraftOf}(x_1, x_0)$, which says if x_0 owns x_1 that can land on x_2 , then x_1 may be the aircraft of the airline x_0 . Evoda also adopts the PCA measure for rule confidence estimation.

3.2 Generate Negative Examples

Some other approaches address the lack of negative examples by generating them, this is particularly necessary for learning rules with negations, traditionally known as *nonmonotonic logic programs* [33, 64]. Such rules can express a form of exceptions and support nonmonotonic reasoning.

Exception-Enriched Rule Learning [21] is an ILP-based method that refines learned Horn rules by adding negated atoms (i.e., exceptions) into their rule bodies. They primarily focus on mining rules over unary predicates, by converting binary predicates into multiple unary ones. For example, the binary predicate `hasOrigin(·, ·)` can be translated into several unary ones like `hasOriginPEK(·)` and `hasOriginBOS(·)`. This is because unary predicates are easier to search for negated atoms. But this makes KG a flattened representation containing just unary facts. To overcome this problem, Nonmonotonic Relational Learning [75] extends [21] to learn exception rules with binary predicates in KGs.

The lack of negative examples is essentially related to the incompleteness and noisiness of KGs, and approaches like RuLES [31] address this through KG embeddings. Compared to AMIE+, RuLES has two more refinement operators to allow negated atoms in a rule body. From Figure 2, we can see the rule evaluation for RuLES is a hybrid combination of statistical and embedding-guided confidence. A weight is used to allow one to choose whether to rely more on the classical measure μ_1 (like standard or PCA confidence), or on the embedding-based measure $\mu_2(\mathcal{G}_r, \phi)$. \mathcal{G}_r extends \mathcal{G} with facts derived from \mathcal{G} by applying rule r . So, $\mu_2(\mathcal{G}_r, \phi)$ capture the information about facts missing in \mathcal{G} that are relevant for r by loss function $\phi(\cdot)$ pre-trained by KG embedding and text corpus models.

4 Statistical Path Generalisation

Statistical Relational Learning (SRL) [26] is a subfield of machine learning that focuses on modelling and learning complex relational data. It combines principles from statistical learning and relational databases to address learning tasks that involve structured data with rich inter-dependencies and uncertainty. There are some SRL approaches to learning first-order rules by using heuristic searching algorithms, for instance, both ProPPR [80, 79] and CoR-PRA [38] are based on the Path Ranking Algorithm (PRA). Several rule learners have emerged to utilise various path-searching strategies based on statistics to directly generate rules. In this section, we focus on sampling-based approaches to mining frequency patterns or path patterns for rule generation in KGs.

4.1 Heuristic Path Sampling

This group of approaches generate candidate rules by directly exploring paths in KGs using heuristic strategies, and these approaches typically focus on learning CP rules generalised from KG paths. As many KGs contain scheme-level (a.k.a., ontological) knowledge such as entity types as well as relation domains and ranges, which is different from instance-level knowledge about relations between individual entities. So, a KG containing ontological knowledge can be split into the ontology graph and the instance (sub)graph.

Given a KG $\mathcal{G} = \mathcal{G}_O \cup \mathcal{G}_I$, \mathcal{G}_O is the *ontology graph* and \mathcal{G}_I is the *instance graph*. The instance graph \mathcal{G}_I has entities as vertices and describes instance-level knowledge about entities and their relations; for instance, a triple (`airline-NY`, `hasBase`, `airport-JFK`) describes that the two entities `airline-NY` and `airport-JFK` are associated by the relation `hasBase`. \mathcal{G}_I also describes the classes of entities, such as a triple (`airline-NY`, `rdf:type`, `Airline`) expressing that `airline-NY` is a member of the class `Airline`. The ontology graph \mathcal{G}_O , on the other hand, has classes as vertices and describes schema-level (or ontological) knowledge about the relations between classes. For example, triple

(Airline, rdfs:subClassOf, Organisation) says that class Airline is a subclass of Organisation. Also, the two triples (hasBase, rdfs:domain, Airline) and (hasBase, rdfs:range, Airport) state that relation hasBase has a domain type Airline and a range type Airport. This can be expressed as an edge (Airline, hasBase, Airport) in the ontology graph.

A path is a sequence of triples $(s_1, p_1, o_1), (s_2, p_2, o_2), \dots, (s_n, p_n, o_n)$ in \mathcal{G} where $o_i = s_{i+1}$ ($1 \leq i < n$). There are two kinds of paths in KGs. A *instance path* (resp., *ontological path*) is a path where s_i, o_i are entities (resp., classes). Existing path exploration methods can thus be classified by whether the paths are from the ontology graph or the instance graph. Generalising instance paths to form rule patterns is also called a *bottom-up* approach, while generating rules from ontological paths is called a *top-down* approach.

4.1.1 Bottom-up Approaches

The AnyBURL series [45, 44, 43] are typical bottom-up approaches, which learn rules by sampling and generalising instance-level paths. To generalise path instances to form rule patterns, AnyBURL substitutes entities from sampled paths in the KG with variables. For example, the paths `worksFor(Allen, airline-NY)`, `hasHeadquater(airline-NY, NewYork)` and `livesIn(Allen, NewYork)` can be generalised to a rule `worksFor(x, y) ∧ hasHeadquater(y, z) → livesIn(x, z)`. Unlike the bottom-up approaches in ILP, the generalisation does not add or delete atoms but only substitutes their parameters. This bottom-up approach allows AnyBURL to learn CP rules with constants (i.e., non-substituted entities) in predefined places, e.g., `worksFor(x, y) ∧ hasHeadquater(y, NewYork) → livesIn(x, NewYork)`. Due to the huge space of semi-grounded CP rules, it is infeasible for AnyBURL to systematically explore the whole search space. Instead, it introduces an anytime algorithm where users can specify the learning time. AnyBURL is much more efficient and effective for rule learning in the sense that it is able to learn more rules in a shorter time than most existing rule learners. For the task of link prediction in KGs, AnyBURL also outperforms many embedding-based methods. In addition, Reinforced AnyBURL[43] is an extension of AnyBURL that introduces reinforcement learning to find more reasonable rules earlier. To achieve this, it incorporates three different reward strategies based on statistical confidence measures to guide the sampling process.

Another example of bottom-up approaches is RuDiK [54]. Different from AnyBURL, it starts from some positive examples and searches the paths connecting the subjects and objects in the positive examples using A^* search. In this approach, the rule generation is guided by assigning a so-called *marginal weight* to each path. In each iteration, the method picks the most promising paths (that is, paths with minimum marginal weight) in the queue of invalid paths. Such paths are expanded and evaluated. If valid, it is added to the output and not further expanded. And the resulting invalid paths will be put back in the queue. This process continues until the queue is empty. RuDiK extends the language bias to allow \perp in the rule heads and literal comparison in the rule bodies. Thus, it can learn constraints as well as rules. To learn rules with literal comparison, it adds comparison edges between pairs of literals of the same type and treats the comparison operators as normal predicates. Unlike AnyBURL using SC for confidence estimation, RuDiK adopts ILP-style example coverage metrics.

4.1.2 Top-down Approaches

Rule learners exploring ontological paths in KGs include Ontological Pathfinding [12], ScaLeKB [13], and RARL [56]. They construct path patterns directly from ontology graphs, and the space of ontological paths is much smaller than that of instance paths. For example, the ontology path `hasBirthPlace(Person, City)`, `isRegionOf(City, Country)` and `hasNationality(Person, Country)` can be generalised to a rule `hasBirthPlace(x0, x1) ∧ isRegionOf(x1, x2) → hasNationality(x0, x2)`.

ScaLeKB extends Ontological Pathfinding and is a typical example of top-down approaches. It achieves its efficiency by storing candidate rules in relational tables according to *structural equivalence*. Two first-order rules are defined to be structurally equivalent if the number of body predicates of the rule and the position of the variable are the same. After that, the rule mining algorithm can process the join queries and partition tables into smaller inputs in parallel, so as to break the mining tasks into smaller independent sub-tasks with Spark.

However, ScaLeKB typically uses type information as hard constraints on the candidate rules, i.e., each entity going through the path must belong to some type. Yet in practice, type information in KGs is often highly incomplete, which would make it too restrictive.

Instead of directly searching ontological paths as hard constraints, RARL [56] samples paths step by step according to the domain and range information of the relations. It reduces the path search space by computing the relatedness of the relations on the paths based on *term frequency-inverse document frequency (TF-IDF)* weighting factor, an information retrieval technique adapted to KGs. RARL adopts ILP-style example coverage metrics similar to RuDiK. It also samples a reduced instance graph for fast confidence computation. Moreover, in TyRuLe [82], type information is present in the learned rules and is encoded as latent representations (known as embeddings) to guide the rule search.

4.2 Frequent Pattern Sampling

Some techniques in association rule mining have been adapted to first-order rule learning [81, 4, 5]. Such techniques are originally developed to discover meaningful relationships or associations among items in a dataset and thus, they are modified to generate first-order rules by identifying statistically frequent patterns occurring in KGs.

RDF2Rules [81] samples the so-called *frequent predicate cycles (FPCs)*. A *predicate cycle* is a sequence of variables and predicates of the form $(x_0, p_1, x_1, \dots, p_n, x_0)$, which are essentially generalised instance paths. If a predicate cycle has a sufficient number of instance paths in the KG as its instantiations, it is called a *frequent predicate cycle*. Rules can be generated from FPCs. For instance, a FPC $(x_0, \text{hasOrigin}^{-1}, x_1, \text{hasDestination}, x_2, \text{hasAlliance}, x_0)$ can generate three rules $\text{hasDestination}(x_1, x_2) \wedge \text{hasAlliance}(x_2, x_0) \rightarrow \text{hasOrigin}(x_1, x_0)$, $\text{hasOrigin}(x_1, x_0) \wedge \text{hasAlliance}^{-1}(x_0, x_2) \rightarrow \text{hasDestination}(x_1, x_2)$, and $\text{hasDestination}^{-1}(x_2, x_1) \wedge \text{hasOrigin}(x_1, x_0) \rightarrow \text{hasAlliance}(x_2, x_0)$. RDF2Rules uses a greedy algorithm to iteratively mine FPCs.

SWARM [4, 5] converts triples in KGs into transaction data to apply association rule mining. Association rules capture frequent items in transaction data. As for KG, they convert a triple (s, p, o) into a 2-tuple $(s, (p, o))$ or $(o, (p^{-1}, s))$. Here both of the s or (p, o) can be seen as items, also the fact tuple $(s, (p, o))$ can be one transaction where these two items appear at the same time. The frequent transaction items can be generalised to rule patterns. For example, we found two frequent transaction items having a common or mostly overlapping item set like $(\{\text{Allen, Ally}\}, (\text{livesIn, NewYork}))$ and $(\{\text{Allen, Ally}\}, (\text{diedIn, NewYork}))$. They can be generalised to the association rule $\{\text{Allen, Ally}\} : (\text{livesIn, NewYork}) \rightarrow (\text{diedIn, NewYork})$. Using the type information like `rdf:type` and `rdfs:subClassOf` in the ontology graph, they could generate the rule $\{\text{Person}\} : (\text{livesIn, City}) \rightarrow (\text{diedIn, City})$.

5 Neuro-symbolic Methods

In this section, we review rule learning methods that utilise deep neural networks (DNNs). While a rich body of DNN-based methods have been proposed for KG completion, including Graph Neural Networks and embedding-based methods, there is an increasing interest in developing or applying DNN-based methods for rule learning over KGs.

There are two major streams of research in this group. Those that learn both rule structures and parameters (i.e., rule plausibility degrees) through DNN models [89, 77, 62, 57, 87, 14, 11], and those that combine DNN models with other rule learning strategies through KG embeddings [88, 51, 94, 82].

5.1 End-to-end Models

In this subsection, we discuss about approaches that use DNNs to learn rules directly by optimising objective functions that roughly correspond to plausible path patterns.

5.1.1 Neural Logic Programming

Neural LP [89] was among the first attempts to combine rule structure learning and confidence learning in an end-to-end differentiable model. It is based on a differentiable probabilistic logic called *TensorLog*, which models CP rule inferences with sparse matrix multiplications. TensorLog maps each entity $e_i \in \mathcal{E}$ to a one-hot vector $\mathbf{v}_i \in \{0, 1\}^{|\mathcal{E}|}$ where only the i -th entry is 1, and each relation $p_k \in \mathcal{P}^*$ to a matrix $\mathbf{M}_k \in \{0, 1\}^{|\mathcal{E}| \times |\mathcal{E}|}$ such that its (i, j) entry is 1 if $p(e_i, e_j)$ is a fact in the KG. Then, the application of a rule $p_1(x, z) \wedge p_2(z, y) \rightarrow p(x, y)$ on an entity $x = e_i$ can be captured by matrix multiplications $\mathbf{M}_1 \cdot \mathbf{M}_2 \cdot \mathbf{v}_i = \mathbf{s}$. The non-zero entries of the score vector \mathbf{s} represent the instances of y (as entities in the KG) when the rule is applied. Hence, the rule-based inference is captured as:

$$\max_{\{\alpha_r, L_r\}} \sum_{\{x, y\}} score(y|x) = \max_{\{\alpha_r, L_r\}} \sum_{\{x, y\}} \mathbf{v}_y^T \left(\sum_r (\alpha_r (\prod_{k \in L_r} \mathbf{M}_k \mathbf{v}_x)) \right), \quad (5)$$

where r indexes over all possible rules, α_r is the confidence associated with rule r and L_r is an ordered list of all predicates in this particular rule. The rule structure along with its confidence α_r are learned by maximising the score. Neural LP thus uses gradient-based programming and optimisation algorithms for the rule learning task. A challenge is a large number of learnable parameters, and Neural LP reduces the learnable parameters by approximating the optimisation objective functions. Another limitation of the above approach is that it is bound to learn rules with a fixed length, and Neural LP addresses this by using an LSTM and attention mechanisms to learn rules of variable lengths.

Neural-Num-LP [77] extends Neural-LP to learn rules with negations and numeric values. It also improves Neural-LP by representing some necessary matrix operations implicitly, including using dynamic programming, cumulative sums operation for numerical comparison features, and low-rank factorisations for negated atoms. Yet it is found that Neural-LP may inevitably learn meaningless rules with high confidence that share atoms with quality rules, and DRUM [62] addresses this issue by utilising bidirectional RNNs to prune the potential incorrect rules.

5.1.2 Decoupling Models

Neural logic programming approaches may still face challenges of exponentially large rule search space, as well as the computational cost of large matrix multiplications. Also, the complexity of simultaneously learning rule structures and confidences makes the optimisation of the computation nontrivial.

To overcome this challenge, new models have been proposed by decoupling the rule structure learning and confidence learning. RNNLogic [57] uses a separate rule generator module for structure learning and a reasoning predictor module for confidence learning. Such a separation allows for more efficient optimisation, by adapting an *Expectation-Maximisation (EM)* algorithm [48], which

enables RNNLogic to handle large KGs effectively. In the same spirit, Ruleformer [87] adopts an encoder-decoder model based on the Transformer architecture. The logic rules are indirectly parsed from the reasoning paths by trained parameters, rather than learned directly. Since Transformer is a sequence-to-sequence model, a converter is used to convert context sub-graph structure into a sequence. A relational attention mechanism is utilised for encoding multi-relational KGs in Transformer. Moreover, RLogic [14] proposes a sequential rule learning algorithm into small atomic models in a recursive way. Given a relation path of rule body $[p_1, \dots, p_n]$, the relation path encoder first reduces it into a single head p_h by recursively merging relation pairs by a greedy algorithm. Then, the close ratio predictor bridges the gap between “ideal prediction” p_h following logical rules and “real observation” p_t given in KGs. A multi-layer perceptron (MLP) is used to learn the probability of replacing a relation pair with a single relation and the ratio that a path will close. Finally, Reinforcement Rule Learning [11] formalises the rule generation problem as a sequential decision problem, by adopting reinforcement learning (RL) to generate rules step by step. As shown in Figure 2, the plausibility of generated candidate rules is estimated with a hybrid measure combining explicit statistical confidence and latent embedding measures.

It should be noted that many DNN-based models aim to learn instance paths in KGs to predict missing triples in link prediction. Logical rules are extracted from these instance paths as the side product [86, 74, 16, 85].

5.2 Embedding-based Methods

Representation learning for KGs has attracted intensive interest, which maps entities, relations, and types to low-dimensional vector or matrix spaces, called *embedding* [61, 35], to capture semantic associations between them. Another stream of rule learning approaches use existing or new KG embeddings to learn rule structures and combine them with other rule search strategies and/or rule confidence measures. Utilising embeddings enhances the efficiency of rule learning to allow the handling of large KGs and improves the interpretability of DNN-based rule learning methods.

The authors of [88] were among the first to suggest using KG embeddings extracted from DNNs for rule structure learning, called EMBEDRULE [88]. In EMBEDRULE, the plausibility of rules will be first estimated via an embedding-based *scoring function* before the more expensive computation of PCA confidence. It first embeds entities $e \in \mathcal{E}$ and predicates $p \in \mathcal{P}^*$ as respectively vectors $\mathbf{e} \in \mathbb{R}^d$ and diagonal matrices $\mathbf{P} \in \mathbb{R}^{d \times d}$. Similar to the bilinear transformation of Neural LP, the embeddings satisfy $\mathbf{e}_1^T \cdot \mathbf{P} \cdot \mathbf{e}_2 \approx 1$ for each fact $p(e_1, e_2)$ in the KG; that is, $\mathbf{e}_1^T \cdot \mathbf{P} \approx \mathbf{e}_2^T$. Consider a CP rule r of the form $p_1(x_0, x_1) \wedge p_2(x_1, x_2) \wedge \dots \wedge p_n(x_{n-1}, x_n) \rightarrow p(x_0, x_n)$, there should be many instance paths that support it, i.e., $p_1(e_0, e_1), p_2(e_1, e_2), \dots, p_n(e_{n-1}, e_n)$ and $p(e_0, e_n)$ in the KG. Hence, the embeddings satisfy $\mathbf{e}_0^T \cdot \mathbf{P}_1 \approx \mathbf{e}_1^T$, $\mathbf{e}_1^T \cdot \mathbf{P}_2 \approx \mathbf{e}_2^T$, \dots , $\mathbf{e}_{n-1}^T \cdot \mathbf{P}_n \approx \mathbf{e}_n^T$, and $\mathbf{e}_0^T \cdot \mathbf{P} \approx \mathbf{e}_n^T$; that is, $\mathbf{e}_0^T \cdot \mathbf{P}_1 \cdot \mathbf{P}_2 \cdot \dots \cdot \mathbf{P}_n \approx \mathbf{e}_n^T \approx \mathbf{e}_0^T \cdot \mathbf{P}$. Since rule r must hold for many such entities e_0 , the rule can be captured by $\mathbf{P}_1 \cdot \mathbf{P}_2 \cdot \dots \cdot \mathbf{P}_n \approx \mathbf{P}$. The scoring function for r is defined via embeddings as follows:

$$\text{score}_1(r) = \text{sim}(\mathbf{P}_1 \cdot \mathbf{P}_2 \cdot \dots \cdot \mathbf{P}_n, \mathbf{P}), \quad (6)$$

and the similarity between two matrices $\text{sim}(\mathbf{M}_1, \mathbf{M}_2)$ can be defined in various ways such as the Frobenius norm, i.e., $\text{sim}(\mathbf{M}_1, \mathbf{M}_2) = \exp(-\|\mathbf{M}_1 - \mathbf{M}_2\|_F)$. Based on similar intuitions, different scoring functions via embeddings for rule learning.

RLvLR [51] uses a KG sampling strategy and path embedding methods, which enables it to handle large-scale KGs like DBpedia or Wikidata. The step of sampling based on n -hop paths (for rules with maximum length n) can effectively reduce the search space and can handle massive benchmarks efficiently. The sampled smaller KG contains only those entities and facts that are relevant to the target predicate p . Then, RLvLR uses the proposed *co-occurrence scoring function*

to guide and prune the search for plausible rules. The selected candidates are kept for the final evaluation by standard confidence and head coverage. The co-occurrence means that for a rule of the form $p_1(x_0, x_1) \wedge p_2(x_1, x_2) \wedge \dots \wedge p_n(x_{n-1}, x_n) \rightarrow p(x_0, x_n)$, the objects of p_i share many common entities with the subjects of p_{i+1} , where embeddings of the subjects and objects of p_i , denoted \mathbf{p}_i^s and \mathbf{p}_i^o , are defined as the averages of the embeddings of the entities occurring in the corresponding positions. Their co-occurrence scoring function is defined as:

$$score_2(r) = sim(\mathbf{p}_1^s, \mathbf{p}^s) + sim(\mathbf{p}_1^o, \mathbf{p}_2^s) + \dots + sim(\mathbf{p}_{n-1}^o, \mathbf{p}_n^s) + sim(\mathbf{p}_n^o, \mathbf{p}^o). \quad (7)$$

Later approaches focus on improving the embedding methods to further enhance rule learning performance [83, 94, 53] or learn more expressive forms of rules [52, 82]. In particular, R-Linker [83] improves RLvLR with a hierarchical sampling and lightweight embedding method, and IterE [94] improves the KG embeddings of (especially sparse) KGs through an iterative enhancement process. Rules are learned from embedding with traverse and select strategies, while embedding is refined according to new triples inferred by rules. StreamLearner [52] extends RLvLR to learn temporal rules, and TyRuLe [82] extends the embedding to learn rules with entity type information. Finally, embeddings have been used to transfer rules from one KG to another KG [53].

6 Applications of Rule Learning

Logical rules have been applied in a wide range of scenarios and play a significant role in Explainable AI. In this section, we focus on the applications of first-order Horn rules automatically learned by rule learners.

6.1 Applications in KG Completion

Many existing KGs are large-scale and subject to regular updates. Yet the knowledge contained in them is still far from complete and contains noise. Manual maintenance of large-scale KGs is costly, if not impossible. Hence, automated reasoning for KG completion and verification is essential, including common tasks such as *link prediction* and *fact checking*. While link prediction aims to discover missing links between entities, fact-checking focuses on validating existing triples. Both tasks are important for building KGs and enhancing the quality of existing KGs.

Rule-based Link Prediction. *Link prediction* is the task of extracting missing triples in a KG and thus it is a subtask of KG completion. Formally, given an entity $s \in \mathcal{E}$ (resp., $o \in \mathcal{E}$) and a relation $p \in \mathcal{P}$ in a KG, the task is to predict entities o (resp., s) such that the triple (s, p, o) is plausibly in the KG.

In recent years, a large number of link prediction models have been proposed. Most major models for link prediction are based on DNNs, especially, embedding models and thus, lack transparency and explainability. Some researchers proposed to develop explainable link prediction models that are based on logic rules [23, 88, 45]. Rules express explicit knowledge and are easy to understand for human beings. For example, given a query of $(\text{Allen}, \text{hasNationality}, ?)$ in Figure 1, the rule $\text{hasBirthPlace}(x, y) \wedge \text{isRegionOf}(y, z) \rightarrow \text{hasNationality}(x, z)$ can predict $\text{hasNationality}(\text{Allen}, \text{America})$ and provide the reason that $\text{hasBirthPlace}(\text{Allen}, \text{Boston}) \wedge \text{isRegionOf}(\text{Boston}, \text{America})$. Thus, rules provide a promising approach to explainable link prediction. This has become feasible when scalable and effective rule learners are available. On the other hand, link prediction offers an important benchmark for validating and evaluating rule learners for KGs. Experiments show that rule-based link prediction methods possess competitive accuracy and scalability compared to embedding-based ones [44, 56].

Unlike embedding-based methods that rank all the entities via scoring functions, a rule-based method derives plausible new facts through logical reasoning, and the confidence of the new facts is determined by the rules deriving them. One research direction is how to calculate the confidence of the derived facts from those of the rules. Noisy-OR [23] and Max-Aggregation [44] are two popular methods for obtaining the confidence degree of a plausible triple that is derived from learned rules. Noisy-OR is defined by aggregating the SC of all the rules deriving the triple. The intuition is that facts inferred by more rules should have a higher confidence degree. Max-Aggregation [44] ranks a triple based on the maximum SC of all rules deriving it. It only considers the most confident rule that derives the triple. These two methods for obtaining confidence degrees have their advantages and disadvantages. When aggregating using Noisy-OR, the redundant rules can lead to overestimating the confidence of predicted entities. In order to mitigate this disadvantage, a Non-redundant Noisy-OR [55] is proposed to cluster rules based on their redundancy degree prior to Noisy-OR. Predictions of rules in a cluster are aggregated using the Max-Aggregation, while predictions of the different clusters are then further aggregated using the Noisy-OR.

Rule-guided Embeddings. Rules can also be used to guide the training of existing embedding models so that logic relations are incorporated into the embeddings. In this way, inferences by the generated embeddings are expected to satisfy the rules and are more interpretable, which essentially combines symbolic rule reasoning with neural networks.

In the early literature of rule-guided embeddings, only a small number of manually created hard rules are used in the models [78, 28]. Given the availability of scalable and effective rule learners for KGs, recent efforts focus on combining embedding models and rules learned automatically. RUGE [29] and SoLE [93] use *t-norm fuzzy logics* [30] to incorporate grounded rules, as the *t-norm fuzzy logics* define how to calculate the probabilities of compositions (e.g., conjunctions and disjunctions) of clauses from the probabilities of the individual clauses. Some other works, such as pLogicNet [58], use *Markov Logic Network (MLN)* [60] to combine the grounded rules with probabilistic graphical models. These approaches need to first compute all the groundings of the rules, which is computationally expensive, especially for long rules. To avoid this, some other approaches incorporate rules as a sequence of relations [17, 27, 50]. Some other models incorporate rules via different learning frameworks, such as RuleGuider [39] and RARL [32], which train reinforcement learning agents guided by rules, and AR-KGAN [95], which incorporates rules via Graph Attention Networks (GAT).

Rule-based Fact Checking. *Fact checking* is the task of verifying the facts in KGs, by predicting the plausibility of the facts. Traditionally, fact checking is done by manual verification which is extremely time-consuming. While black-box models can be used to estimate the plausibility of facts, rule-based fact checking offers explainability. Moreover, rules contain domain-specific knowledge, which is particularly useful for verifying ambiguous facts.

Some rule-based fact checking approaches can generate evidence or explanations for the facts under examination (called target facts) to assess their plausibility [20, 67, 22]. CHEEP [20] generates evidence as paths in the ontology graph according to rules, while ExFaKT [22] uses rules to rewrite each target fact into a set of other easier-to-spot facts as explanations from both text and KGs.

There are also some approaches dedicated to designing new forms of rules for fact checking. Lin et al. [40] propose a kind of graph-fact-checking (GFC) rules to discover a discriminant-directed graph associated with the target facts. OGFC [41] extends GFC with more topological and ontological information to group similar triples. Rules in disjunctive normal form (DNF), i.e., a disjunction of multiple conjunctions, are used in CHAI [9] to filter facts. Some other

methods transform rule-based fact-checking problems into answer set program formulations, like EXPCLAIM [1]. It uses first-order rule discovery and Web text mining to gather the evidence to assess target facts, and the fact-checking task is modelled as an inference problem in the answer set programs.

6.2 Other Applications of Rule Learning

There has been a heightened interest in the interpretability of AI models, as understanding how and why a model arrives at a particular decision is pivotal for trust and transparency in decision-making. Logic rules, with their inherent ability to support human-comprehensible reasoning processes, have emerged as a valuable tool in diverse fields.

Several domains are actively exploring the use of automatically learned rules to complement and enhance existing AI models. Rules can naturally be applied to other tasks related to KGs, such as entity alignment [10, 36] and knowledge base question answering (KBQA) [70]. In the entity alignment task, RTEA-RA [36] enhances the embeddings of individual entities by injecting the grounded rules into the model to produce hybrid embeddings. MuGNN [10] reconciles the structural differences of two KGs before entity alignment by employing rules induced by AMIE+ for KG completion and pruning. Also, these rules are transferred between KGs based on the knowledge invariant assumption.

As for KBQA, RuKBC-QA [70] uses rule-based knowledge base completion (KBC) in general question answering (QA) systems. Both the origin knowledge base and inferred missing facts by selected rules are used as input of RuKBC-QA for predicting the answers.

Beyond the tasks with KGs as the primary forms of data, several attempts involving rules have been made in the context of natural language processing (NLP), computer vision (CV), and biomedical applications. For NLP, RuleBERT [63] tries to teach the pre-trained language models (PLMs) with the common-sense knowledge provided by Horn rules. KoRC [91] uses rules learned by background KGs to construct the reasoning chain for Reading Comprehension.

Rule learning has also found applications in computer vision, like reasoning in the sub-graph extracted from the images. LOGICDEF [90] constructs a defence model that uses first-order logic rules mined from the extracted scene graph, to explain the object classification and detect the attacks of the adversarial vision model. The document image model [59] uses inductive rules by extracting textual subgraphs corresponding to the text entities in the documents for information extraction.

In the biomedical domain, rule learning has also shown promising applications. In drug discovery [66], generating the explanation paths by Horn rules for drug-disease (entity) pairs. As for drug-gene interaction prediction [2], rules can predict missing links between drug and gene nodes in a graph that contains relevant biomedical knowledge.

These studies collectively showcase the versatility and potential of rule learning across diverse domains, addressing various challenges and improving outcomes in different application areas.

7 Discussion and Conclusion

Rule learning in KGs is a fast-developing research area with promising applications in various fields. In this section, we discuss some future research directions for rule learning and conclude the paper.

7.1 Future Directions

Despite the rapid advancements in rule learning techniques over KGs, this is still a relatively new research area, with a lot of potential for further development in several aspects, including the complexity and quality of learned rules, evaluation metrics, and rule learning and reasoning strategies.

First of all, the complexity of learned rules can be enhanced by expanding the form of rules (i.e., the language bias) to be learned. Most existing rule learning approaches focus on rules that represent path patterns in KGs. While path patterns capture important structural information in KGs, more complex structural patterns such as sub-graph patterns can reveal more refined knowledge and first-order Horn rules can express complex sub-graph patterns. Scalability of rule learning and identification of the most useful sub-graph patterns remain major challenges in learning more complex rules. Besides, learning rules that can capture more useful information such as attributes or more complex logical connections in KGs is another aspect of rule complexity.

Also, the form of rules and their learning strategies can be tailored according to the applications. For instance, to learn domain-specific rules that are designed to express prior knowledge in specific domains, can potentially lead to more scalable, accurate, and robust performance in the concerned applications. This is especially relevant for applications in knowledge-intensive domains such as biomedical science, healthcare, finance, or legal applications, which involve domain-specific knowledge and specialised applications.

Moreover, to enhance the quality of learned rules, it is crucial to establish effective and robust rule quality evaluation measures. Existing rule confidence measures often lack sufficient granularity or interpretability. Yet designing suitable rule quality metrics has received less attention in the literature. Meanwhile, there has been insufficient emphasis on evaluating the semantic validity of learned rules, that is, to measure how meaningful the learned rules are to human beings. This is particularly important for learning domain-specific rules, and it is desirable to develop quality measures that take into consideration both the data semantics and data distributions.

Finally, new rule learning and reasoning strategies that tightly integrate symbolic AI and deep learning techniques hold great promise. This line of research may involve developing hybrid models that leverage the strengths of both rule-based reasoning and data-driven neural networks. Yet a tighter integration of them has always been a pursuit of the academic communities and the industry. With the recent development of neuro-symbolic approaches, where neural networks are used for symbolic rule learning and reasoning, the other direction is gaining increasing interest, that is, to apply symbolic knowledge, logical constraints, and rule guidance in neural network predictions. This research direction also foresees a tighter coupling of rule learning and prediction models that utilise KGs.

7.2 Conclusion

In this paper, we have provided a systematic review of state-of-the-art rule learning for knowledge graphs. We studied major categories of logic rule learning approaches over knowledge graphs, including the ILP-based, statistical path generalisation, and neuro-symbolic approaches, with discussions on their developments and limitations. Besides, we also discussed the rule confidence measures, which play a crucial role in rule learning. As for the applications of rule learning, we introduced applications of rule-based knowledge graph inferences, as well as wider applications in natural language processing, computer vision, and biomedical science. Finally, we pointed out several promising future directions for rule learning research.

References

- 1 Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. Explainable fact checking with probabilistic answer set programming. In *Proceedings of the 2019 Truth and Trust Online Conference (TTO-19), London, UK, October 4-5, 2019*, 2019. URL: https://truthandtrustonline.com/wp-content/uploads/2019/09/paper_15.pdf.
- 2 Fotis Aisopos and Georgios Paliouras. Comparing methods for drug-gene interaction prediction on the biomedical literature knowledge graph: performance versus explainability. *BMC Bioinform.*, 24(1):272, 2023. doi:10.1186/S12859-023-05373-2.
- 3 Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference (ISWC-07), Busan, Korea, November 11-15, 2007*, volume 4825, pages 722–735. Springer, 2007. doi:10.1007/978-3-540-76298-0_52.
- 4 Molood Barati, Quan Bai, and Qing Liu. SWARM: an approach for mining semantic association rules from semantic web data. In *Proceedings of the 14th Pacific Rim International Conference on Artificial Intelligence (PRICAI-16), Phuket, Thailand, August 22-26, 2016*, volume 9810, pages 30–43. Springer, 2016. doi:10.1007/978-3-319-42911-3_3.
- 5 Molood Barati, Quan Bai, and Qing Liu. Mining semantic association rules from RDF data. *Knowl. Based Syst.*, 133:183–196, 2017. doi:10.1016/J.KNSYS.2017.07.009.
- 6 Luigi Bellomarini, Davide Benedetto, Georg Gottlob, and Emanuel Sallinger. Vadalog: A modern architecture for automated reasoning with large knowledge graphs. *Inf. Syst.*, 105:101528, 2022. doi:10.1016/J.IS.2020.101528.
- 7 Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. doi:10.1038/scientificamerican0501-3.
- 8 Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-08), Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM, 2008. doi:10.1145/1376616.1376746.
- 9 Agustín Borrego, Daniel Ayala, Inma Hernández, Carlos R. Rivero, and David Ruiz. Generating rules to filter candidate triples for their correctness checking by knowledge graph completion techniques. In *Proceedings of the 10th International Conference on Knowledge Capture (KCAP-19), Marina Del Rey, CA, USA, November 19-21, 2019*, pages 115–122. ACM, 2019. doi:10.1145/3360901.3364418.
- 10 Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL-19), Florence, Italy, July 28- August 2, 2019*, pages 1452–1461. Association for Computational Linguistics, 2019. doi:10.18653/V1/P19-1140.
- 11 Lihan Chen, Sihang Jiang, Jingping Liu, Chao Wang, Sheng Zhang, Chenhao Xie, Jiaqing Liang, Yanghua Xiao, and Rui Song. Rule mining over knowledge graphs via reinforcement learning. *Knowl. Based Syst.*, 242:108371, 2022. doi:10.1016/J.KNSYS.2022.108371.
- 12 Yang Chen, Sean Goldberg, Daisy Zhe Wang, and Soumitra Siddharth Johri. Ontological pathfinding. In *Proceedings of the 2016 International Conference on Management of Data Conference (SIGMOD-16), San Francisco, CA, USA, June 26 -July 01, 2016*, pages 835–846. ACM, 2016. doi:10.1145/2882903.2882954.
- 13 Yang Chen, Daisy Zhe Wang, and Sean Goldberg. Scalekb: scalable learning and inference over large knowledge bases. *VLDB J.*, 25(6):893–918, 2016. doi:10.1007/S00778-016-0444-3.
- 14 Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. Rlogic: Recursive logical rule learning from knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-22), Washington, DC, USA, August 14 - 18, 2022*, pages 179–189. ACM, 2022. doi:10.1145/3534678.3539421.
- 15 Andrew Cropper, Sebastijan Dumancic, Richard Evans, and Stephen H. Muggleton. Inductive logic programming at 30. *Mach. Learn.*, 111(1):147–172, 2022. doi:10.1007/S10994-021-06089-1.
- 16 David Jaime Tena Cucala, Bernardo Cuenca Grau, Egor V. Kostylev, and Boris Motik. Explainable gnn-based models over knowledge graphs. In *Proceedings of the 10th International Conference on Learning Representations (ICLR-22), Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL: <https://openreview.net/forum?id=CrCvGNHAIrz>.
- 17 Boyang Ding, Quan Wang, Bin Wang, and Li Guo. Improving knowledge graph embedding using simple constraints. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL-18), Melbourne, Australia, July 15-20, 2018*, pages 110–121. Association for Computational Linguistics, 2018. doi:10.18653/V1/P18-1011.
- 18 Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmman, Shaohua Sun, and Wei Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-14), New York, USA, August 24-27, 2014*, pages 601–610. ACM, 2014. doi:10.1145/2623330.2623623.
- 19 Facebook. Graph api - facebook for developers. <https://developers.facebook.com/docs/graph-api/>, 2022.
- 20 Valeria Fionda and Giuseppe Pirrò. Fact checking via evidence patterns. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18), July 13-19, 2018, Stock-*

- holm, Sweden, pages 3755–3761. ijcai.org, 2018. doi:10.24963/IJCAI.2018/522.
- 21 Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. Exception-enriched rule learning from knowledge graphs. In *Proceedings of the 15th International Semantic Web Conference (ISWC-16), Kobe, Japan, October 17-21, 2016*, pages 234–251, 2016. doi:10.1007/978-3-319-46523-4_15.
 - 22 Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. Exfakt: A framework for explaining facts over knowledge graphs and text. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM-19), Melbourne, VIC, Australia, February 11-15, 2019*, pages 87–95. ACM, 2019. doi:10.1145/3289600.3290996.
 - 23 Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB J.*, 24(6):707–730, 2015. doi:10.1007/S00778-015-0394-1.
 - 24 Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd International World Wide Web Conference (WWW-13), Rio de Janeiro, Brazil, May 13-17, 2013*, pages 413–422. International World Wide Web Conferences Steering Committee / ACM, 2013. doi:10.1145/2488388.2488425.
 - 25 Matt Gardner, Partha P. Talukdar, and Tom Michael Mitchell. Combining vector space embeddings with symbolic logical inference over open-domain text. In *Proceedings of the 2015 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 22-25, 2015*. AAAI Press, 2015. URL: <http://www.aaai.org/ocs/index.php/SSS/SSS15/paper/view/10230>.
 - 26 Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. MIT press, 2007.
 - 27 Shu Guo, Lin Li, Zhen Hui, Lingshuai Meng, Bingnan Ma, Wei Liu, Lihong Wang, Haibin Zhai, and Hong Zhang. Knowledge graph embedding preserving soft logical regularity. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM-20), Virtual Event, Ireland, October 19-23, 2020*, pages 425–434. ACM, 2020. doi:10.1145/3340531.3412055.
 - 28 Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16), Austin, Texas, USA, November 1-4, 2016*, pages 192–202. The Association for Computational Linguistics, 2016. doi:10.18653/V1/D16-1019.
 - 29 Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4816–4823. AAAI Press, 2018. doi:10.1609/AAAI.V32I1.11918.
 - 30 Petr Hájek. *Metamathematics of Fuzzy Logic*, volume 4. Kluwer, 1998. doi:10.1007/978-94-011-5300-3.
 - 31 Vinh Thinh Ho, Daria Stepanova, Mohamed H. Gad-Elrab, Evgeny Kharlamov, and Gerhard Weikum. Rule learning from knowledge graphs guided by embedding models. In *Proceedings of the 17th International Semantic Web Conference (ISWC-18), Monterey, CA, USA, October 8-12, 2018*, volume 11136, pages 72–90. Springer, 2018. doi:10.1007/978-3-030-00671-6_5.
 - 32 Zhongni Hou, Xiaolong Jin, Zixuan Li, and Long Bai. Rule-aware reinforcement learning for knowledge graph reasoning. In *Findings of the Association for Computational Linguistics (ACL/IJCNLP-21), Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021, pages 4687–4692. Association for Computational Linguistics, 2021. doi:10.18653/V1/2021.FINDINGS-ACL.412.
 - 33 Katsumi Inoue and Yoshimitsu Kudoh. Learning extended logic programs. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan, August 23-29, 1997, 2 Volumes*, pages 176–181. Morgan Kaufmann, 1997. URL: <http://ijcai.org/Proceedings/97-1/Papers/029.pdf>.
 - 34 Lei Ji. Knowledge mining api - microsoft research. <https://www.microsoft.com/en-us/research/project/knowledge-mining-api/>, 2016.
 - 35 Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514, 2022. doi:10.1109/TNNLS.2021.3070843.
 - 36 Tingting Jiang, Chenyang Bu, Yi Zhu, and Xindong Wu. Two-stage entity alignment: Combining hybrid knowledge graph embedding with similarity-based relation alignment. In *Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence (PRICAI-19), Cuvu, Yanuca Island, Fiji, August 26-30, 2019*, volume 11670, pages 162–175. Springer, 2019. doi:10.1007/978-3-030-29908-8_13.
 - 37 Jonathan Lajus, Luis Galárraga, and Fabian M. Suchanek. Fast and exact rule mining with AMIE 3. In *Proceedings of the 17th Extended Semantic Web Conference (ESWC-20), Heraklion, Crete, Greece, May 31-June 4, 2020*, volume 12123, pages 36–52. Springer, 2020. doi:10.1007/978-3-030-49461-2_3.
 - 38 Ni Lao, Einat Minkov, and William W. Cohen. Learning relational features with backward random walks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-15), July 26-31, 2015, Beijing, China*, pages 666–675. The Association for Computer Linguistics, 2015. doi:10.3115/V1/P15-1065.
 - 39 Deren Lei, Gangrong Jiang, Xiaotao Gu, Kexuan Sun, Yuning Mao, and Xiang Ren. Learning collaborative agents with rule guidance for knowledge graph reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP-20), Online, November 16-20, 2020*, pages 8541–8547. Associ-

- ation for Computational Linguistics, 2020. doi:10.18653/V1/2020.EMNLP-MAIN.688.
- 40 Peng Lin, Qi Song, Jialiang Shen, and Yinghui Wu. Discovering graph patterns for fact checking in knowledge graphs. In *Proceedings of the 23rd International Conference on Database Systems for Advanced Applications (DASFAA-18), Gold Coast, QLD, Australia, May 21-24, 2018*, volume 10827, pages 783–801. Springer, 2018. doi:10.1007/978-3-319-91452-7_50.
 - 41 Peng Lin, Qi Song, and Yinghui Wu. Fact checking in knowledge graphs with ontological subgraph patterns. *Data Sci. Eng.*, 3(4):341–358, 2018. doi:10.1007/S41019-018-0082-4.
 - 42 Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI-22), Virtual Event, February 22 - March 1, 2022*, pages 4120–4127. AAAI Press, 2022. doi:10.1609/AAAI.V36I4.20330.
 - 43 Christian Meilicke, Melisachew Wudage Chekol, Manuel Fink, and Heiner Stuckenschmidt. Reinforced anytime bottom up rule learning for knowledge graph completion. *CoRR*, abs/2004.04412, 2020. doi:10.48550/arXiv.2004.04412.
 - 44 Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, August 10-16, 2019*, pages 3137–3143. ijcai.org, 2019. doi:10.24963/IJCAI.2019/435.
 - 45 Christian Meilicke, Manuel Fink, Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt. Fine-grained evaluation of rule- and embedding-based systems for knowledge graph completion. In *Proceedings of the 17th International Semantic Web Conference (ISWC-18), Monterey, CA, USA, October 8-12, 2018*, volume 11136, pages 3–20. Springer, 2018. doi:10.1007/978-3-030-00671-6_1.
 - 46 George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995. doi:10.1145/219717.219748.
 - 47 Stephen H. Muggleton. Inductive logic programming. In *Proceedings of the 1st International Workshop on Algorithmic Learning Theory (ALT-90), Tokyo, Japan, October 8-10, 1990*, pages 42–62. Springer/Ohmsha, 1990.
 - 48 Radford M. Neal and Geoffrey E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, volume 89, pages 355–368. Springer Netherlands, 1998. doi:10.1007/978-94-011-5014-9_12.
 - 49 Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *IEEE*, 104(no. 1):11–33, 2016. doi:10.1109/JPROC.2015.2483592.
 - 50 Guanglin Niu, Yongfei Zhang, Bo Li, Peng Cui, Si Liu, Jingyang Li, and Xiaowei Zhang. Rule-guided compositional representation learning on knowledge graphs. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, February 7-12, 2020*, pages 2950–2958. AAAI Press, 2020. doi:10.1609/AAAI.V34I03.5687.
 - 51 Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. Scalable rule learning via learning representation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18), July 13-19, 2018, Stockholm, Sweden*, pages 2149–2155. ijcai.org, 2018. doi:10.24963/IJCAI.2018/297.
 - 52 Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. Learning temporal rules from knowledge graph streams. In *Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE-19), Stanford University, Palo Alto, California, USA, March 25-27, 2019*, volume 2350. CEUR-WS.org, 2019. URL: <https://ceur-ws.org/Vol-2350/paper15.pdf>.
 - 53 Pouya Ghiasnezhad Omran, Zhe Wang, and Kewen Wang. Knowledge graph rule mining via transfer learning. In *Proceedings of the 23th Pacific-Asia Knowledge Discovery and Data Mining Conference (PAKDD-19), Macau, China, April 14-17, 2019*, volume 11441, pages 489–500. Springer, 2019. doi:10.1007/978-3-030-16142-2_38.
 - 54 Stefano Ortona, Venkata Vamsikrishna Meduri, and Paolo Papotti. Rudik: Rule discovery in knowledge bases. *PVLDB*, 11(12):1946–1949, 2018. doi:10.14778/3229863.3236231.
 - 55 Simon Ott, Christian Meilicke, and Matthias Samwald. SAFRAN: an interpretable, rule-based link prediction method outperforming embedding models. In *Proceedings of the 3rd Conference on Automated Knowledge Base Construction (AKBC-21), Virtual, October 4-8, 2021*, 2021. doi:10.24432/C5MK57.
 - 56 Giuseppe Pirrò. Relatedness and tbox-driven rule learning in large knowledge bases. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, February 7-12, 2020*, pages 2975–2982. AAAI Press, 2020. doi:10.1609/AAAI.V34I03.5690.
 - 57 Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *Proceedings of the 9th International Conference on Learning Representations (ICLR-21), Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: <https://openreview.net/forum?id=tGZu6D1breV>.
 - 58 Meng Qu and Jian Tang. Probabilistic logic neural networks for reasoning. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems 2019 (NeurIPS-19), December 8-14, 2019, Vancouver, BC, Canada*, pages 7710–7720, 2019. doi:10.48550/arXiv.1906.08495.
 - 59 Mouli Rastogi, Syed Afshan Ali, Mrinal Rawat, Lovekesh Vig, Puneet Agarwal, Gautam Shroff, and Ashwin Srinivasan. Information extraction from document images via FCA based template detection and knowledge graph rule induction.

- In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW-20)*, Seattle, WA, USA, June 14-19, 2020, pages 2377-2385. Computer Vision Foundation / IEEE, 2020. doi:10.1109/CVPRW50498.2020.00287.
- 60 Matthew Richardson and Pedro M. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107-136, 2006. doi:10.1007/S10994-006-5833-1.
 - 61 Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Trans. Knowl. Discov. Data*, 15(2):14:1-14:49, 2021. doi:10.1145/3424672.
 - 62 Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. DRUM: end-to-end differentiable rule mining on knowledge graphs. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS-19)*, December 8-14, 2019, Vancouver, BC, Canada, pages 15321-15331, 2019. URL: <https://proceedings.neurips.cc/paper/2019/hash/0c72cb7ee1512f800abe27823a792d03-Abstract.html>.
 - 63 Mohammed Saeed, Naser Ahmadi, Preslav Nakov, and Paolo Papotti. Rulebert: Teaching soft rules to pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP-21)*, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 1460-1476. Association for Computational Linguistics, 2021. doi:10.18653/v1/2021.EMNLP-MAIN.110.
 - 64 Chiaki Sakama. Induction from answer sets in non-monotonic logic programs. *ACM Trans. Comput. Log.*, 6(2):203-231, 2005. doi:10.1145/1055686.1055687.
 - 65 Abdus Salam, Rolf Schwitter, and Mehmet A. Orgun. Probabilistic rule learning systems: A survey. *ACM Comput. Surv.*, 54(4):79:1-79:16, 2022. doi:10.1145/3447581.
 - 66 Kara Schatz, Cleber C. Melo-Filho, Alexander Tropsha, and Rada Chirkova. Explaining drug-discovery hypotheses using knowledge-graph patterns. In *Proceedings of the IEEE International Conference on Big Data, Orlando, FL, USA, December 15-18, 2021*, pages 3709-3716. IEEE, 2021. doi:10.1109/BIGDATA52589.2021.9672006.
 - 67 Baoxu Shi and Tim Weninger. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowl. Based Syst.*, 104:123-133, 2016. doi:10.1016/J.KNSYS.2016.04.015.
 - 68 Ashwin Srinivasan. The aleph manual, 2007. URL: https://www.cs.ox.ac.uk/activities/programinduction/Aleph/aleph_toc.html.
 - 69 Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web (WWW-07)*, Banff, Alberta, Canada, May 8-12, 2007, pages 697-706. ACM, 2007. doi:10.1145/1242572.1242667.
 - 70 Qilin Sun and Weizhuo Li. Rukbc-qa: A framework for question answering over incomplete kbs enhanced with rules injection. In *Proceedings of the 9th Natural Language Processing and Chinese Computing (NLPC-20)*, Zhengzhou, China, October 14-18, 2020, volume 12431, pages 82-94. Springer, 2020. doi:10.1007/978-3-030-60457-8_7.
 - 71 Martin Svatos, Steven Schockaert, Jesse Davis, and Ondrej Kuzelka. Strike: Rule-driven relational learning using stratified k-entailment. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI-20)*, 29 August-8 September 2020, Santiago de Compostela, Spain, volume 325, pages 1515-1522. IOS Press, 2020. doi:10.3233/FAIA200259.
 - 72 Thomas Pellissier Tanon, Daria Stepanova, Simon Razniewski, Paramita Mirza, and Gerhard Weikum. Completeness-aware rule learning from knowledge graphs. In *Proceedings of the 16th International Semantic Web Conference (ISWC-17)*, Vienna, Austria, October 21-25, 2017, volume 10587, pages 507-525. Springer, 2017. doi:10.1007/978-3-319-68288-4_30.
 - 73 Thomas Pellissier Tanon, Daria Stepanova, Simon Razniewski, Paramita Mirza, and Gerhard Weikum. Completeness-aware rule learning from knowledge graphs. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18)*, July 13-19, 2018, Stockholm, Sweden, pages 5339-5343. ijcai.org, 2018. doi:10.24963/IJCAI.2018/749.
 - 74 Komal K. Teru, Etienne G. Denis, and William L. Hamilton. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning (ICML-20)*, 13-18 July 2020, Virtual Event, volume 119, pages 9448-9457. PMLR, 2020. URL: <http://proceedings.mlr.press/v119/teru20a.html>.
 - 75 Hai Dang Tran, Daria Stepanova, Mohamed H. Gad-Elrab, Francesca A. Lisi, and Gerhard Weikum. Towards nonmonotonic relational learning from knowledge graphs. In *Proceedings of the 26th International Conference on Inductive Logic Programming (ILP-16)*, London, UK, September 4-6, 2016, volume 10326, pages 94-107. Springer, 2016. doi:10.1007/978-3-319-63342-8_8.
 - 76 Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78-85, 2014. doi:10.1145/2629489.
 - 77 Po-Wei Wang, Daria Stepanova, Csaba Domokos, and J. Zico Kolter. Differentiable learning of numerical rules in knowledge graphs. In *Proceedings of the 8th International Conference on Learning Representations (ICLR-20)*, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL: <https://openreview.net/forum?id=rJleKgrKwS>.
 - 78 Quan Wang, Bin Wang, and Li Guo. Knowledge base completion using embeddings and rules. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI-15)*, Buenos Aires, Argentina, July 25-31, 2015, pages 1859-1866. AAAI Press, 2015. URL: <http://ijcai.org/Abstract/15/264>.
 - 79 William Yang Wang and William W. Cohen. Joint information extraction and reasoning: A scalable statistical relational learning approach. In *Proceedings of the 53rd Annual Meeting of the Asso-*

- ciation for Computational Linguistics (ACL-15), July 26-31, 2015, Beijing, China, pages 355–364. The Association for Computer Linguistics, 2015. doi:10.3115/V1/P15-1035.
- 80 William Yang Wang, Kathryn Mazaitis, and William W. Cohen. Structure learning via parameter learning. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM-14)*, Shanghai, China, November 3-7, 2014, pages 1199–1208. ACM, 2014. doi:10.1145/2661829.2662022.
- 81 Zhichun Wang and Juan-Zi Li. Rdf2rules: Learning rules from RDF knowledge bases by mining frequent predicate cycles. *CoRR*, abs/1512.07734, 2015. doi:10.48550/arXiv.1512.07734.
- 82 Hong Wu, Zhe Wang, Kewen Wang, and Yi-Dong Shen. Learning typed rules over knowledge graphs. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR-22)*, Haifa, Israel, July 31-August 5, 2022, 2022. doi:10.24963/kr.2022/51.
- 83 Hong Wu, Zhe Wang, Xiaowang Zhang, Pouya Ghiasnezhad Omran, Zhiyong Feng, and Kewen Wang. A system for reasoning-based link prediction in large knowledge graphs. In *Proceedings of the 18th International Semantic Web Conference Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) (ISWC-19)*, Auckland, New Zealand, October 26-30, 2019, volume 2456, pages 121–124. CEUR-WS.org, 2019. URL: <https://ceur-ws.org/Vol-2456/paper32.pdf>.
- 84 Lianlong Wu, Emanuel Sallinger, Evgeny Sherkhonov, Sahar Vahdati, and Georg Gottlob. Rule learning over knowledge graphs with genetic logic programming. In *Proceedings of the 38th IEEE International Conference on Data Engineering (ICDE-22)*, Kuala Lumpur, Malaysia, May 9-12, 2022, pages 3373–3385. IEEE, 2022. doi:10.1109/ICDE53745.2022.00318.
- 85 Yi Xia, Junyong Luo, Mingjing Lan, Gang Zhou, Zhibo Li, and Shuo Liu. Reason more like human: Incorporating meta information into hierarchical reinforcement learning for knowledge graph reasoning. *Appl. Intell.*, 53(11):13293–13308, 2023. doi:10.1007/S10489-022-04147-2.
- 86 Wenhan Xiong, Thien Hoang, and William Yang Wang. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP-17)*, Copenhagen, Denmark, September 9-11, 2017, pages 564–573. Association for Computational Linguistics, 2017. doi:10.18653/V1/D17-1060.
- 87 Zezhong Xu, Peng Ye, Hui Chen, Meng Zhao, Huajun Chen, and Wen Zhang. Ruleformer: Context-aware rule mining over knowledge graph. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING-22)*, Gyeongju, Republic of Korea, October 12-17, 2022, pages 2551–2560. International Committee on Computational Linguistics, 2022. URL: <https://aclanthology.org/2022.coling-1.225>.
- 88 Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR-2015)*, San Diego, CA, USA, May 7-9, 2015, 2015. URL: <http://arxiv.org/abs/1412.6575>.
- 89 Fan Yang, Zhilin Yang, and William W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS-17)*, December 4-9, 2017, Long Beach, CA, USA, pages 2319–2328, 2017. URL: <https://proceedings.neurips.cc/paper/2017/hash/0e55666a4ad822e0e34299df3591d979-Abstract.html>.
- 90 Yuan Yang, James Clayton Kerce, and Faramarz Fekri. LOGICDEF: an interpretable defense framework against adversarial examples via inductive scene graph reasoning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI-22)*, Virtual Event, February 22 - March 1, 2022, pages 8840–8848. AAAI Press, 2022. doi:10.1609/AAAI.V36I8.20865.
- 91 Zijun Yao, Yantao Liu, Xin Lv, Shulin Cao, Jifan Yu, Juanzi Li, and Lei Hou. Korc: Knowledge oriented reading comprehension benchmark for deep text understanding. In *Findings of the Association for Computational Linguistics (ACL-23)*, Toronto, Canada, July 9-14, 2023, pages 11689–11707. Association for Computational Linguistics, 2023. doi:10.18653/V1/2023.FINDINGS-ACL.743.
- 92 Qiang Zeng, Jignesh M. Patel, and David Page. Quickfoil: Scalable inductive logic programming. *Proc. VLDB Endow.*, 8(3):197–208, 2014. doi:10.14778/2735508.2735510.
- 93 Jindou Zhang and Jing Li. Enhanced knowledge graph embedding by jointly learning soft rules and facts. *Algorithms*, 12(12):265, 2019. doi:10.3390/A12120265.
- 94 Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. Iteratively learning embeddings and rules for knowledge graph reasoning. In *Proceedings of the World Wide Web Conference (WWW-19)*, San Francisco, CA, USA, May 13-17, 2019, pages 2366–2377. ACM, 2019. doi:10.1145/3308558.3313612.
- 95 Zhenghao Zhang, Jianbin Huang, and Qinglin Tan. Association rules enhanced knowledge graph attention network. *Knowl. Based Syst.*, 239:108038, 2022. doi:10.1016/J.KNOSYS.2021.108038.