Rule Learning from Knowledge Graphs Guided by Embedding Models

V. Thinh Ho[1] D. Stepanova[1] M. Gad-Elrab[1] E. Kharlamov[2] G. Weikum[1]

[1] Max Planck Institute for Informatics [2] University of Oxford

Abstract. Rules over a Knowledge Graph (KG) capture interpretable patterns in data and various methods for rule learning have been proposed. Since KGs are inherently incomplete, rules can be used to deduce missing facts. Statistical measures for learned rules such as confidence reflect rule quality well when the KG is reasonably complete; however, these measures might be misleading otherwise. So it is difficult to learn high-quality rules from the KG alone, and scalability dictates that only a small set of candidate rules is generated. Therefore, the ranking and pruning of candidate rules is a major problem. To address this issue, we propose a rule learning method that utilizes probabilistic representations of missing facts. In particular, we iteratively extend rules induced from a KG by relying on feedback from a precomputed embedding model over the KG and external information sources including text corpora. Experiments on real-world KGs demonstrate the effectiveness of our novel approach both with respect to the quality of the learned rules and fact predictions that they produce.

1 Introduction

Motivation. Rules are widely used to represent relationships and dependencies between data items in datasets and to capture the underlying patterns in data [2,27]. Applications of rules include health-care [40], telecommunications [21], manufacturing [3], and commerce [30,18]. In order to facilitate rule construction, a variety of rule learning methods have been developed, see e.g. [10,19] for an overview. Moreover, various statistical measures such as confidence, actionability, and unexpectedness to evaluate the *quality* of the learned rules have been proposed.

Rule learning has recently been adapted to the setting of Knowledge Graphs (KGs) [12,38,11,34] where data is represented as a graph of entities interconnected via relations and labeled with classes, or more formally as a set of grounded binary and unary atoms typically referred to as facts. Examples of large-scale KGs include Wikidata [35], Yago [32], NELL [23], and Google's KG [1]. Since many KGs are constructed from semi-structured knowledge, such as Wikipedia, or harvested from the Web with a combination of statistical and linguistic methods, they are inherently incomplete [26,12].

Rules over KGs are of the form $head \leftarrow body$, where $head$ is a binary atom and $body$ is a conjunction of, possibly negated, binary or unary atoms. When rules are automatically learned, statistical measures like support and confidence are used to assess the quality of rules. Most notably, the confidence of a rule is the fraction of facts predicted by the rule that are indeed true in the KG. However, this is a meaningful measure for rule quality only when the KG is reasonably complete. For rules learned from largely incomplete KGs, confidence and other measures may be misleading, as they do not reflect the patterns in the missing facts. For example, a KG that knows only (or mostly) male CEOs would yield a heavily biased rule $\mathit{gender}(X, \mathit{male}) \leftarrow \mathit{isCEO}(X, Y), \mathit{isCompany}(Y)$, which does not extend to the entirety of valid facts beyond the KG. Therefore, it is crucial that rules can be ranked by a meaningful quality measure, which accounts for KG incompleteness.

Example. Consider a KG about people's jobs, residence and spouses as well as office locations and headquarters of companies. Suppose a rule learning method has computed the following two rules:

$$r_1 : \mathit{livesIn}(X, Y) \leftarrow \mathit{worksFor}(X, Z), \mathit{hasOfficeIn}(Z, Y) \tag{1}$$
$$r_2 : \mathit{livesIn}(Y, Z) \leftarrow \mathit{marriedTo}(X, Y), \mathit{livesIn}(X, Z) \tag{2}$$

The rule r_1 is quite noisy, as companies have offices in many cities, but employees live and work in only one of them, while the rule r_2 clearly is of higher quality. However, depending on how the KG is populated with instances, the rule r_1 could nevertheless score higher than r_1 in terms of confidence measures. For example, the KG may contain only a specific subset of company offices and only people who work for specific companies. If we knew the complete KG, then the rule r_2 should presumably be ranked higher than r_1.

Suppose we had a perfect oracle for the true and complete KG. Then we could learn even more sophisticated rules such as

$$r_3 : \mathit{livesIn}(X, Y) \leftarrow \mathit{worksFor}(X, Z), \mathit{hasHeadquarterIn}(Z, Y), \mathit{not\ locatedIn}(Y, \mathit{USA}).$$

This rule would capture that most people work in the same city as their employers' headquarters, with the USA being an exception (assuming that people there are used to long commutes). This is an example of a rule that contains a negated atom in the rule body (so it is no longer a Horn rule) and has a partially grounded atom with a variable and a constant as its arguments.

Problem. The problem of KG incompleteness has been tackled by methods that (learn to) predict missing facts for KGs (or actually missing relational edges between existing entities). A prominent class of approaches is statistics-based and includes tensor factorization, e.g. [25] and neural-embedding-based models, e.g. [4,24]. Intuitively, these approaches turn a KG, possibly augmented with external sources such as text [39,41], into a probabilistic representation of its entities and relations, known as *embeddings*, and then predict the likelihood of missing facts by reasoning over the embeddings (see, e.g. [36] for a survey).

These kinds of embeddings can complement the given KG and are a potential asset in overcoming the limitations that arise from incomplete KGs. Consider the following gedankenexperiment: we compute embeddings from the KG and external text sources, that can then be used to predict the complete KG that comprises all valid facts. This would seemingly be the perfect starting point for learning rules, without the bias and quality problems of the incomplete KG. However, this scenario is way oversimplified. The embeddings-based fact predictions would themselves be very noisy, yielding also many spurious facts. Moreover, the computation of all fact predictions and the induction of all possible rules would come with a big scalability challenge: in practice, we need to restrict ourselves to computing merely small subsets of likely fact predictions and promising rule candidates.

Approach. In this work we propose an approach for rule learning guided by external sources that allows to learn high-quality rules from incomplete KGs. In particular, our method extends rule learning by exploiting probabilistic representations of missing facts computed by embedding models of KGs and possibly other external information sources. We iteratively construct rules over a KG and collect feedback from a precomputed embedding model, through specific queries issued to the model for assessing the quality of (partially constructed) rule candidates. This way, the rule induction loop is interleaved with the guidance from the embeddings, and we avoid scalability problems. Our machinery is also more expressive than many prior works on rule learning from KGs, by allowing non-monotonic rules with negated atoms as well as partially grounded atoms. Within this framework, we devise confidence measures that capture rule quality better than previous techniques and thus improve the ranking of rules.

Contribution. The salient contributions of our work are as follows.

– We propose a rule learning approach guided by external sources, and show how to learn high-quality rules, utilizing feedback from embedding models.
– We implement our approach and present extensive experiments on real-world KGs, demonstrating the effectiveness of our approach with respect to both the quality of the learned rules and the fact predictions that they produce.
– Our code and data are made available to the research community at `http://people.mpi-inf.mpg.de/~gadelrab/RuLES/`

2 Rule Learning Guided by External Sources

In this section we first give some necessary preliminaries, then we introduce our framework for rule learning guided by external sources, discuss challenges associated with it, and finally propose a concrete instantiation of our framework with embedding models.

2.1 Background

We assume countable sets $\mathcal{R}$ of unary and binary relation names and $\mathcal{C}$ of constants. A *knowledge graph* (KG) $\mathcal{G}$ is a finite set of ground atoms a of the form $P(b, c)$ and $C(b)$ over $\mathcal{R} \cup \mathcal{C}$. With $\Sigma_{\mathcal{G}}$, the *signature* of $\mathcal{G}$, we denote elements of $\mathcal{R} \cup \mathcal{C}$ that occur in $\mathcal{G}$.

We define rules over KGs following the standard approach of non-monotonic logic programs under the answer set semantics [13]. Let $\mathcal{X}$ be a countable set of variables. A *rule* r is of the form $head \leftarrow body$, where $head$, or $head(r)$, is an atom over $\mathcal{R} \cup \mathcal{C} \cup \mathcal{X}$ and $body$, or $body(r)$, is a conjunction of positive and negative atoms over $\mathcal{R} \cup \mathcal{C} \cup \mathcal{X}$. Finally, $body^+(r)$ and $body^-(r)$ denote the atoms that occur in $body(r)$ positively and negatively respectively, that is, the rule can be written as $head(r) \leftarrow body^+(r), not\ body^-(r)$. A rule is *Horn*, if all head variables occur in the body, and $body^-(r)$ is empty.

We define *execution* of rules with default negation [13] over KGs in the standard way. More precisely, let $\mathcal{G}$ be a KG, r a rule over $\Sigma_{\mathcal{G}}$, and a be an atom over $\Sigma_{\mathcal{G}}$. Then, $r \models_{\mathcal{G}} a$ holds if there is a variable assignment that maps atoms $body^+(r)$ in $\mathcal{G}$ such that it does not map any of the atoms in $body^-(r)$ in $\mathcal{G}$. Then, let $\mathcal{G}_r = \mathcal{G} \cup \{a \mid r \models_{\mathcal{G}} a\}$. Intuitively, $\mathcal{G}_r$ extends $\mathcal{G}$ with edges derived from $\mathcal{G}$ by applying r.

2.2 Problem Statement and Proposal of General Solution

Let $\mathcal{G}$ be a KG over the signature $\Sigma_{\mathcal{G}} = (\mathcal{R}_{\mathcal{G}}, \mathcal{C}_{\mathcal{G}})$. A *probabilistic KG* $\mathcal{P}$ is a pair $\mathcal{P} = (\mathcal{G}, f)$ where $f : \mathcal{R}_{\mathcal{G}} \times \mathcal{C}_{\mathcal{G}} \times \mathcal{C}_{\mathcal{G}} \to [0, 1]$ is a probability function over the facts over $\Sigma_{\mathcal{G}}$ such that for each atom $a \in \mathcal{G}$ it holds that $f(a) = 1$.

The goal of our work is to learn rules that not only describe the available graph $\mathcal{G}$ well, but also predict highly probable facts based on the function f. The key questions now are how to define the quality of a given rule r based on $\mathcal{P}$ and how to exploit this quality during rule learning for pruning out not promising rules.

A quality measure μ for rules over probabilistic KGs is a function $\mu : (r, \mathcal{P}) \mapsto \alpha$, where $\alpha \in [0, 1]$. In order to measure the quality μ of r over $\mathcal{P}$ we propose:

- to measure the quality μ_1 of r over $\mathcal{G}$, where $\mu_1 : (r, \mathcal{G}) \mapsto \alpha \in [0, 1]$, and
- to measure the quality μ_2 of $\mathcal{G}_r$ by relying on $\mathcal{P}_r = (\mathcal{G}_r, f)$, where $\mu_2 : (\mathcal{G}', (\mathcal{G}, f)) \mapsto \alpha \in [0, 1]$ for $\mathcal{G}' \supseteq \mathcal{G}$ is the quality of extensions $\mathcal{G}'$ of $\mathcal{G}$ over $\Sigma_{\mathcal{G}}$ given f, and
- to combine the result as the weighted sum.

That is, we define our hybrid rule quality function $\mu(r, \mathcal{P})$ as follows:

$$\mu(r, \mathcal{P}) = (1 - \lambda) \times \mu_1(r, \mathcal{G}) + \lambda \times \mu_2(\mathcal{G}_r, \mathcal{P}). \tag{3}$$

In this formula μ_1 can be any classical quality measure of rules over complete graphs. Intuitively, $\mu_2(\mathcal{G}_r, \mathcal{P})$ is the quality of $\mathcal{G}_r$ wrt f that allows us to capture the information about facts missing in $\mathcal{G}$ that are relevant for r. The weighting factor λ, we call it *embedding weight*, allows one to choose whether to rely more on the classical measure μ_1 or on the measure μ_2 of the quality of the facts that are predicted by r over $\mathcal{G}$.

Challenges. There are several challenges that one faces when realising our approach. First, given an incomplete $\mathcal{G}$, one has to define f such that $(\mathcal{G}, f)$ satisfies the expectations, i.e., reflects well the probabilities of missing facts. Second, one has to define μ_1 and μ_2 that also satisfy the expectations and admit efficient implementation. Finally, the adaptation of existing rule learning approaches to account for the probabilistic function f without the loss of scalability is not trivial. Indeed, materializing f by augmenting $\mathcal{G}$ with all possible probabilistic facts over $\Sigma_{\mathcal{G}}$ and subsequently applying standard rule learning methods on the obtained graph is not practical. Storing such potentially enormous augmented graph where many probabilistic facts are irrelevant for the extraction of meaningful rules might be simply infeasible.

2.3 Realization of General Solution

We now describe how we addressed the above stated challenges. In Section 2.3 we present concrete realizations of f, μ_1 and μ_2, and in Section 3 we discuss how we implemented them and adapted within an end-to-end rule learning system.

Realization of the probabilistic function f. We propose to define f by relying on embeddings of KGs. Embeddings are low-dimensional vector spaces that represent nodes and edges of KGs and can be used to estimate the likelihood (not necessary probability) of potentially missing binary atoms using a scoring function $\xi : \mathcal{R} \times \mathcal{C} \times \mathcal{C} \to \rm I\!R$. Examples of concrete scoring functions can be found, e.g., in [36]. Since embeddings per se are

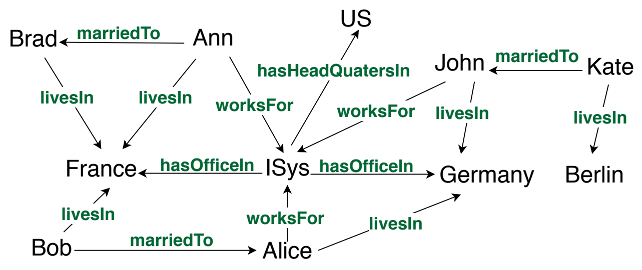


Fig. 1: An example of a knowledge graph.

not in the focus of our paper, we will not give further details on them and refer the reader to [36] for an overview. Note that our framework is not dependent on a concrete embedding model. What is important for us is that embeddings can be used to construct probabilistic representations [24] of atoms missing in KGs and we use this to define f.

Consider an auxiliary definition. Given a KG $\mathcal{G}$, and an atom $a = p(s, o)$, the set $\mathcal{G}_s$ consists of a and all atoms a' that are obtained from a by replacing s with a constant from $\Sigma_{\mathcal{G}}$, except for those that are already in $\mathcal{G}$. Then, given a scoring function ξ, $[\mathcal{G}_s]$ is a list of atoms from $\mathcal{G}_s$ ordered in the descending order. Finally, the *subject rank* [14] of a given ξ, $subject\_rank_\xi(a)$ is the position of a in $[\mathcal{G}_s]$. Analogously one can define $[\mathcal{G}_o]$ and the corresponding *object rank* [14] of a given ξ, that is, $object\_rank_\xi(a)$.

Now we are ready to define the function f for an atom a as the average of its subject and object inverted ranks given ξ [14], i.e.:

$$f_\xi(a) = 0.5 \times (1/subject\_rank_\xi(a) + 1/object\_rank_\xi(a)).$$

Realization of μ_1. This measure should reflect the descriptive quality of a given rule r with respect to $\mathcal{G}$. There are many classical data mining measures that can be used as μ_1, see, e.g. [22,12,33,44] for μ_1s proposed specifically for KGs.

In our work we selected the following two measures for μ_1: *confidence* and *PCA confidence* [12], where PCA stands for the partial completeness assumption, that can be defined using rule support, *r-supp*, partial rule support, *pr-supp*, and body support, *b-supp* as follows. Let $r:\ head \leftarrow body^+, not\ body^-$ be a rule and x a variable occurring in $head$, and let h denote a variable assignment that we with a slight abuse of notation use as a homomorphism on (sets of) atoms. Then,

$$\begin{aligned}
r\text{-}supp(r, \mathcal{G}) &:= |\{h \mid h(head) \in \mathcal{G}, \exists h' \supseteq h \text{ s.t. } h'(body^+) \in \mathcal{G}, h'(body^-) \notin \mathcal{G}\}|,\\
pr\text{-}supp(r, \mathcal{G}) &:= |\{h \mid h(head) \notin \mathcal{G}, \exists h' \text{ s.t. } h(x) = h'(x),\ h'(head) \in \mathcal{G}, \text{ and}\\
&\qquad h(body^+) \in \mathcal{G}, h(body^-) \notin \mathcal{G}\}|,\\
b\text{-}supp(r, \mathcal{G}) &:= |\{h \mid h(body^+) \in \mathcal{G}, h(body^-) \notin \mathcal{G}\}|.
\end{aligned}$$

Finally, we are ready to define confidence and PCA confidence:

$$\begin{aligned}
conf(r, \mathcal{G}) &:= r\text{-}supp(r, \mathcal{G})/b\text{-}supp(r, \mathcal{G}),\\
conf_{pca}(r, \mathcal{G}) &:= r\text{-}supp(r, \mathcal{G})/pr\text{-}supp(r, \mathcal{G}).
\end{aligned}$$

Intuitively, confidence of a rule is the conditional probability of rule's head given its body, while PCA confidence is its generalisation to the open world assumption (OWA), which does not penalize rules that predict facts $p(s, o)$, such that $p(s, o') \notin \mathcal{G}$ for any o'.

Example 1. Consider the KG $\mathcal{G}$ in Fig. 1 and recall the rules r_1 and r_2 from Equations (1)-(2). For r_1, we have $\mathit{conf}(r_1, \mathcal{G}) = \mathit{conf}_{pca}(r_1, \mathcal{G}) = \frac{3}{6}$, while for r_2 it holds that $\mathit{conf}(r_2, \mathcal{G}) = \mathit{conf}_{pca}(r_2, \mathcal{G}) = \frac{1}{3}$. If Alice was not known to live in Germany, then $\mathit{conf}_{pca}(r_2, \mathcal{G}) = \frac{1}{2}$. Finally, for the following rule with negation

$$r_4 : \mathit{livesIn}(Y, Z) \leftarrow \mathit{marriedTo}(X, Y), \mathit{livesIn}(X, Z), \mathit{not}\ \mathit{researcher}(X),$$

that states that married people live together unless one is a researcher, and $\mathcal{G}' = \mathcal{G} \cup \{\mathit{researcher}(bob)\}$, we have $\mathit{conf}(r_4, \mathcal{G}') = \mathit{conf}_{pca}(r_4, \mathcal{G}') = \frac{1}{2}$. □

Realization of μ_2. There are various ways how one can define the quality $\mu_2(\mathcal{G}_r, \mathcal{P})$ of $\mathcal{G}_r$. A natural candidate to define the quality of $\mathcal{G}_r$ is the probability of $\mathcal{G}_r$, that is, as $\mu_2(\mathcal{G}_r, \mathcal{P}) = \prod_{a \in \mathcal{G}_r} f(a) \times \prod_{a \in (\mathcal{R}_\mathcal{G} \times \mathcal{C}_\mathcal{G} \times \mathcal{C}_\mathcal{G}) \setminus \mathcal{G}_r} (1 - f(a))$. A disadvantage of such quality measure is that in practice it will be very low, as the product of many (potentially) small probabilities, and thus Equation 3 will be heavily dominated by $\mu_1(r, \mathcal{G})$. Therefore, we advocate to define $\mu_2(\mathcal{G}_r, \mathcal{P})$ as the average probability of a fact in $\mathcal{G}_r$:

$$\mu_2(\mathcal{G}_r, \mathcal{P}) = (\Sigma_{a \in \mathcal{G}_r \setminus \mathcal{G}} f(a)) / |\mathcal{G}_r \setminus \mathcal{G}|.$$

Example 2. Consider the KG $\mathcal{G}$ in Figure 1, and the rules from Equations (1)-(2) with their confidence values as presented in Example 1. Suppose that a text-enhanced embedding model produced a relatively accurate estimation of the probabilities of facts over $\mathit{livesIn}$ relation. For example, even though within the graph there is no direct connection between Germany and Berlin, relying on the living places of entities similar to John and hidden semantic relations between Germany and Berlin such as co-occurrences in text and other linguistic features, for the fact $a = \mathit{livesIn}(john, berlin)$ we obtained $f(a) = 0.9$, while for $a' = \mathit{livesIn}(john, france)$, a much lower probability $f(a') = 0.09$. These naturally support the predictions of r_2 but not those of r_1.

Generalising this idea, assume that on the whole dataset we get $\mu_2(\mathcal{G}_{r_1}, \mathcal{P}) = 0.1$ and $\mu_2(\mathcal{G}_{r_2}, \mathcal{P}) = 0.8$, where $\mathcal{P} = (\mathcal{G}, f)$. Thus, for $\lambda = 0.5$ we have $\mu(r_1, \mathcal{P}) = (1-0.5) \times 0.5 + 0.5 \times 0.1 = 0.3$, while for $\mu(r_2, \mathcal{P}) = (1-0.5) \times \frac{1}{3} + 0.5 \times 0.8 \approx 0.57$, resulting in the desired ranking of r_2 over r_1 based on μ. □

3 System Implementation

In this section we describe our end-to-end rule learning system with embedding support. Conceptually, our system generalizes the standard relational association rule learners [12,15] to account for the feedback from the probabilistic function f. Following common practice [12] we restrict ourselves to rules that are *closed*, where every variable appears at least twice, and *safe*, where the Horn part is closed.

Overview. The input of the system are a KG, possibly a text corpus, and a set of user specified parameters that are used to terminate rule construction. These parameters include

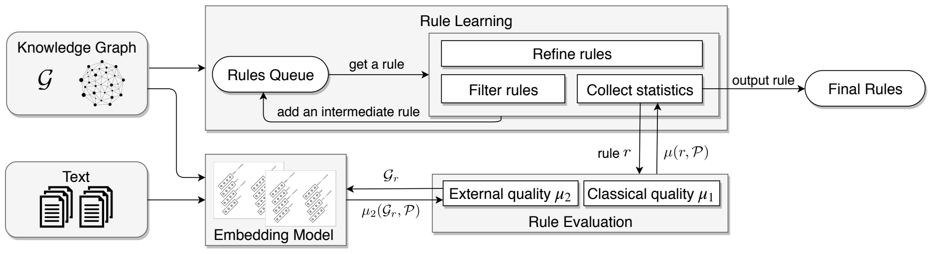


Fig. 2: Overview of our system.

an embedding weight λ, a minimum threshold for μ_1, a minimum rule support *r-supp* and other *rule-related* parameters such as a maximum number of positive and negative atoms allowed in $body(r)$. The KG and text corpus are used to train the embedding model that in turn is used to construct the probabilistic function f. The rules r are constructed in the iterative fashion, starting from the head, by adding atoms to its body one after another until at least one of the termination criteria (that depend on f) is met. In parallel with the construction of r the quality $\mu(r)$ is computed.

In Figure 2 we present a high level architecture of our system, where arrows depict information flow between blocks. The *Rule Learning* block constructs rules over the input KG, *Rule Evaluation* supplies it with quality scores μ for rules r, using $\mathcal{G}$ and f, where f is computed by the *Embedding Model* block from $\mathcal{G}$ and text.

We now discuss the algorithm behind the *Rule Learning* block in Figure 2. Following [12] we model rules as sequences of atoms, where the first atom is the head of the rule and other atoms are its body. The algorithm maintains a priority queue of intermediate rules (see the *Rules Queue* block in Figure 2). Initially all possible binary atoms appearing in $\mathcal{G}$ are added to the queue with empty bodies. At each iteration, a single rule is selected from the queue. If the rule satisfies the *filtering criteria* (see the *Filer rules* block) which we define below, then the system returns it as an output. If the rule is not filtered, then it is processed with one of the *refinement operators* (see the *Refine rules* block) that we define below that expand the rule with one more atom and produce new rule candidates, which are then pushed into the queue (if not being pushed before). The iterative process is repeated until the queue is empty. All the reported rules will be finally ranked by the decreasing order of the hybrid measure μ, computed in *Collect statistics* block.

In the remainder of the section we discuss refinement operators and filtering criteria.

Refinement operators. We rely on the following three standard refinement operators [12] that extend rules:

(*i*) *add a positive dangling atom*: add a new positive atom with one fresh variable and another one appearing in the rule, i.e., *shared*.
(*ii*) *add a positive instantiated atom*: add a positive atom with one argument being a constant and the other one being a shared variable.
(*iii*) *add a positive closing atom*: add a positive atom with both of its arguments being shared variables.

Additionally, we introduce two more operators to allow negated atoms in rule bodies:

(iv) *add an exception instantiated atom*: add a negated atom with one of its arguments being a constant, and the other one being a shared variable.
(v) *add an exception closing atom*: add a binary negated atom to the rule with both of its arguments being shared variables or a unary negated atom with a shared variable.

These two operators are only applied to closed rules to ensure the safety condition. Moreover, we also require the satisfaction of the following condition: for a rule $r : head(r) \leftarrow body^{+}(r)$ the addition of an exception atom should result in the rule $r' : head(r) \leftarrow body^{+}(r), body^{-}(r)$, such that $r\text{-}supp(r, \mathcal{G}) = r\text{-}supp(r', \mathcal{G})$. Intuitively, the application of exception refinement operator must not lead to the decrease of the rule support, i.e., exceptions should explain the absence of predictions expected to be in the graph rather then their presence.

Filtering criteria. After applying one of the refinement operators to a rule, a set of candidate rules is obtained. For each candidate rule we first verify that the hybrid measure μ has increased and discard the rule if it does not. Then, we compute its *h-cover* [12] and our novel exception confidence measure *e-conf* that are defined as follows:

$$\begin{aligned} h\text{-}cover(r, \mathcal{G}) &:= r\text{-}supp(r, \mathcal{G}) / |\{h \mid h(head(r, \mathcal{G})) \in \mathcal{G}\}|, \\ e\text{-}conf(r, \mathcal{G}) &:= conf(r', \mathcal{G}), \end{aligned}$$

where $r' : body^{-}(r) \leftarrow body^{+}(r), not\ head(r)$. If the *h-cover* and *e-conf* are below the user specified threshold, then the rule is discarded. Intuitively, *h-cover* quantifies the ratio of the known true facts that are implied by the rule. In contrast, *e-conf* is the conditional probability of the exception given predictions produced by the Horn part of r, which helps to disregard insignificant exceptions, i.e., those that explain the absence in $\mathcal{G}$ of only a small fraction of predictions made by $head(r) \leftarrow body^{+}(r)$, as such exceptions likely correspond to noise. Observe that not all of the filtering criteria are relevant for all rule types. For example, exception confidence is relevant only for non-monotonic rules to ensure the quality of the added exceptions.

Finally, note that by exploiting the embedding feedback, we can now distinguish exceptions from noise. Consider the rule stating that married people live together. This rule can have several possible exceptions, e.g., either one of the spouses is a researcher or he/she works at a company, which has headquarter in the US. Whenever the rule is enriched with an exception, naturally, the support of its body decreases, i.e., the size of $\mathcal{G}_r$ goes down. Ideally, we want to add such negated atoms, that the average quality of $\mathcal{G}_r$ increases, as this will witness that by adding negated atoms to the rule we get rid of unlikely predictions.

4 Evaluation

We have implemented our hybrid rule learning approach in Java within a system prototype RuLES, and conducted experiments on a Linux machine with 80 cores and 500GB RAM. In this section we report the results of our experimental evaluation, which focuses on *(i)* the benefits of our hybrid embedding-based rule quality measure over traditional rule measures; *(ii)* the effectiveness of RuLES against the state-of-art Horn rule learning systems; and *(iii)* the quality of non-monotonic rules learned by RuLES compared to existing methods.

4.1 Experimental Setup

Datasets. We performed experiments on the following two real world datasets:

- *FB15K* [4]: a subset of Freebase with 592K binary facts over 15K entities and 1345 relations commonly used for evaluating KG embedding models [36].
- *Wiki44K*: a dataset with 250K binary facts over 44K entities and 100 relations, which is a subset of Wikidata dataset from December 2014 used in [12].

In the experiments for each incomplete KG $\mathcal{G}$ we need its *ideal* completion $\mathcal{G}^i$ that would give us a gold standard for evaluating our approach and comparing it to others. Since obtaining a real life $\mathcal{G}^i$ is hard, we used the KGs FB15K and Wiki44K as reference graphs $\mathcal{G}^i_{appr}$ that approximate $\mathcal{G}^i$. We then constructed $\mathcal{G}$ by randomly selecting 80% of its facts while preserving the distribution of facts over predicates.

Embedding models. We experimented with the three state-of-the-art embedding models: TransE [4], HolE [24], and the text-enhanced SSP [41] model. We reuse the implementation of TransE, HolE[1], and SSP[2]. TransE and HolE were trained on $\mathcal{G}$ and SSP on $\mathcal{G}$ enriched with a textual description for each entity extracted from Wikidata. We compared the effectiveness of the models and selected for every KG the best one. Apart from SSP, which showed the best performance on both KGs, we also selected HolE for FB15K and TransE for Wiki44K. Note that in this work as a proof of concept we considered some of the most popular embedding models, but conceptually any model (see [36] for overview) can be used in our system.

Evaluation metric. To evaluate the learned rules we use the quality of predictions that they produce when applied on $\mathcal{G}$, i.e., the more correct facts beyond $\mathcal{G}$ a ruleset produces, the better it is. We consider two evaluation settings: *closed world* setting (CW) and *open world* setting (OW). In the CW setting, we define the prediction precision of a rule r and a set of rules R as

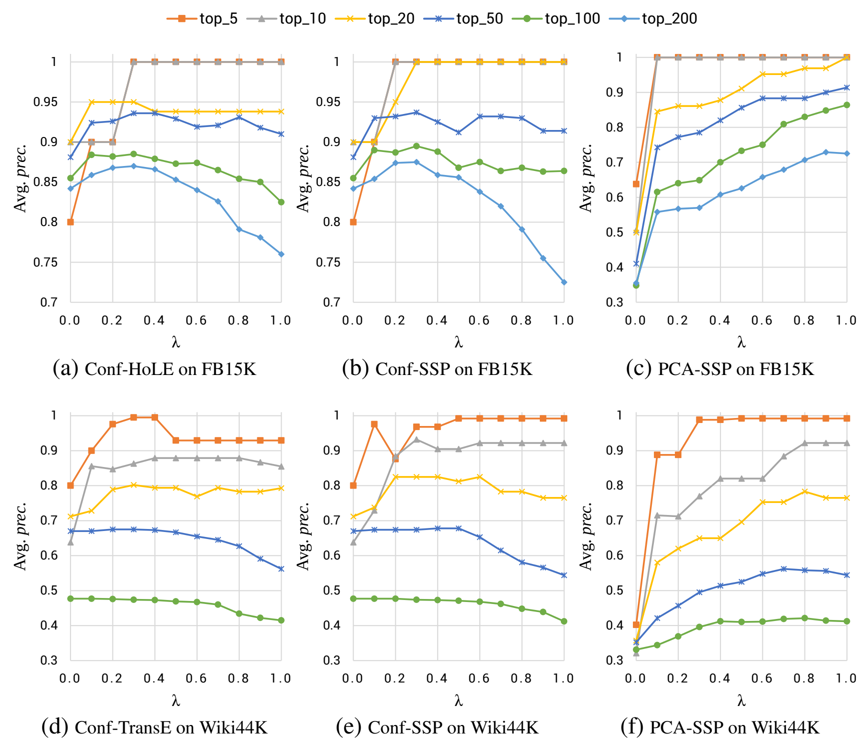
$$pred\_prec_{CW}(r) = \frac{|\mathcal{G}_r \cap \mathcal{G}^i_{appr} \setminus \mathcal{G}|}{|\mathcal{G}_r \setminus \mathcal{G}|}, \quad pred\_prec_{CW}(R) = \frac{\sum\limits_{r \in R} pred\_prec_{CW}(r)}{|R|}.$$

In the OW setting, we also take into account the incompleteness of $\mathcal{G}^i_{appr}$ and consider the quality of predictions outside it by performing a random sampling and manually annotating the sampled facts relying on Web resources such as Wikipedia. Thus, we define the OW prediction precision $pred\_prec_{OW}$ for a set of rules R as follows:

$$pred\_prec_{OW}(R) = \frac{|\mathcal{G}' \cap \mathcal{G}^i_{appr}| + |\mathcal{G}' \backslash \mathcal{G}^i_{appr}| \times accuracy(\mathcal{G}' \backslash \mathcal{G}^i_{appr})}{|\mathcal{G}'|},$$

where $\mathcal{G}' = \bigcup_{r \in R} \mathcal{G}_r \backslash \mathcal{G}$ is the union of predictions generated by rules in R, and $accuracy(S)$ is the approximated ratio of true facts inside S computed via manual checking of facts sampled from S. For simplicity, $Prec.$ is used in tables to refer to

[1] https://github.com/mnick/scikit-kge
[2] https://github.com/bookmanhan/Embedding



(a) Conf-HoLE on FB15K (b) Conf-SSP on FB15K (c) PCA-SSP on FB15K

(d) Conf-TransE on Wiki44K (e) Conf-SSP on Wiki44K (f) PCA-SSP on Wiki44K

Fig. 3: Avg. prediction precision of the *top-k* rules with various *embedding weights*.

$pred\_prec_{OW}$. Finally, to evaluate the meaningfulness of exceptions in a rule (i.e., negated atoms) we compute the *revision precision*, which according to [34] is defined as the ratio of incorrect facts in the difference between predictions produced by the Horn part of a rule and its non-monotonic version over the total number of predictions in this difference (the higher the revision precision, the better the rule exceptions) computed per ruleset. Formally,

$$rev\_prec_{OW}(R) = 1 - \frac{|\mathcal{G}'' \cap \mathcal{G}^i_{appr}| + |\mathcal{G}'' \backslash \mathcal{G}^i_{appr}| \times accuracy(\mathcal{G}'' \backslash \mathcal{G}^i_{appr})}{|\mathcal{G}''|},$$

where $\mathcal{G}'' = \mathcal{G}_H \backslash \mathcal{G}_R$ and H is the set of Horn parts of rules in R. Intuitively, $\mathcal{G}''$ contains facts not predicted by the rules in R but predicted by their Horn versions.

RuLES configuration. We run RuLES in several configurations where μ_1 is set to either *standard confidence (Conf)* or *PCA confidence (PCA)*, and μ_2 is computed based on either TransE, HolE, or SSP models. Through the experiments the configurations are named as μ_1-μ_2 (*e.g.*, Conf-HolE).

4.2 Embedding-Based Hybrid Quality Function

In this experiment we study the effect of using our hybrid embedding-based rule measure μ from Equation 3 on the rule ranking compared to traditional measures. We do it

	FB15K				**Wiki44K**			
top-k	**Conf**	**PCA**	**Conf-HolE**	**Conf-SSP**	**Conf**	**PCA**	**Conf-TransE**	**Conf-SSP**
	$(\lambda = 0)$	$(\lambda = 0)$	$(\lambda = 0.3)$	$(\lambda = 0.3)$	$(\lambda = 0)$	$(\lambda = 0)$	$(\lambda = 0.3)$	$(\lambda = 0.3)$
5	0.800	0.638	**1.000**	**1.000**	0.800	0.402	**0.995**	0.968
10	0.900	0.506	**1.000**	**1.000**	0.638	0.321	0.863	**0.932**
20	0.900	0.499	0.950	**1.000**	0.712	0.357	0.802	**0.825**
50	0.881	0.410	0.936	**0.937**	0.670	0.352	**0.675**	0.674
100	0.855	0.348	0.885	**0.895**	**0.477**	0.331	0.474	0.474
200	0.842	0.355	0.870	**0.875**	–	–	–	–

Table 1: Avg. prediction precision of the rules learned using different measures.

	FB15K						**Wiki44K**					
top-k	**AMIE-PCA**		**AMIE-Conf**		**RuLES**		**AMIE-PCA**		**AMIE-Conf**		**RuLES**	
	Facts	*Prec.*	*Facts*	*Prec.*	*Facts*	*Prec.*	*Facts*	*Prec.*	*Facts*	*Prec.*	*Facts*	*Prec.*
20	1029	0.28	82	0.63	44	1.00	185	0.73	91	0.95	3291	0.98
50	1716	0.43	190	0.74	186	0.92	47099	0.10	3594	0.95	6154	0.88
100	3085	0.65	255	0.78	539	0.80	56831	0.20	13870	0.83	13253	0.82
200	10586	0.62	1210	0.83	1205	0.88	82288	0.39	19538	0.72	20408	0.73
500	40050	0.51	2702	0.75	7124	0.95	219264	0.35	124836	0.23	128256	0.48

Table 2: Prediction precision of the *top-k* rules generated by RuLES and AMIE.

by first learning rules of the form $r:\ \ h(X,Z) \leftarrow p(X,Y), q(Y,Z)$ from $\mathcal{G}$ where $\textit{r-supp}(r,\mathcal{G}) \geq \mathit{10}$, $\mathit{conf}(r,\mathcal{G}) \in [\mathit{0.1}, \mathit{1})$ and $\textit{h-cover}(r,\mathcal{G}) \geq \mathit{0.01}$. Then, we rank these rules using Equation 3 with $\lambda \in \{0, 0.1, 0.2, \ldots, 1\}$, $\mu_1 \in \{\mathit{conf}, \mathit{conf}_{pca}\}$ and with μ_2 that is computed by relying on TransE, HolE and SSP. Note that $\lambda = 0$ corresponds to the standard rule measure μ_1 alone.

Figure 3 shows the average prediction precision $\mathit{pred\_prec}_{CW}$ of the *top-k* rules ranked using our measure μ for different embedding weights λ (*x axis*). In particular, in Figures 3a, 3b, 3d, and 3e we observe that combining confidence with any embedding model results in the increase in the average prediction precision for $0 \leq \lambda \leq 0.3$. Moreover, we observe the decrease of prediction precision for $0.4 \leq \lambda \leq 1$ and *top-k* rules learned from FB15K when $k \geq 20$ and from Wiki44K when $k \geq 10$. This shows that the combination of μ_1 and μ_2 gives noticeable positive effect on the prediction results. On the other hand, for $\mu_1 = \mathit{conf}_{pca}$ the precision increases significantly when combined with embedding models and only decreases slightly for $\lambda = 1$ (Figures 3c,3f). Utilizing conf_{pca} instead of conf as μ_1 in our hybrid measure is less effective, since our training data $\mathcal{G}$ is randomly sampled breaking the partial completeness assumption adopted by the PCA confidence.

Table 1 compactly summarizes the average prediction precision of *top-k* rules ranked by the standard rule measures and our μ for the best value of $\lambda = 0.3$ and highlights the effect of using the better embedding model (text-enhanced vs standard). We observe that the accuracy of a utilized embedding model is naturally propagated to the accuracy of the rules that we obtain using our hybrid ranking measure μ. This demonstrates that the use of a better embedding model positively effects the quality of learned rules.

top-k	**Family**			
	NeuralLP		**Conf-TransE**	
	Facts	*Prec.*	*Facts*	*Prec.*
10	3709	0.72	4201	0.68
20	8821	0.53	6957	0.72
30	11337	0.49	9368	0.71
40	14662	0.46	11502	0.72
50	18768	0.40	14547	0.62

Table 3: Prediction precision of the *top-k* rules generated by NeuralLP and RuLES.

r^1: $nationality(X, Y) \leftarrow graduated\_from(X, Z), in\_country(Z, Y),$ **not** $research\_uni(Z)$
r^2: $scriptwriter\_of(X, Y) \leftarrow preceded\_by(X, Z), scriptwriter\_of(Z, Y),$ **not** $tv\_series(Z)$
r^3: $noble\_family(X, Y) \leftarrow spouse(X, Z), noble\_family(Z, Y),$ **not** $chinese\_dynasties(Y)$

Table 4: Example rules with exception generated by RuLES.

4.3 Horn Rule Learning

In this experiment, we compare RuLES under Conf-SSP configuration (with embedding weight $\lambda = 0.3$) with the state-of-art Horn rule learning system AMIE. We used the default AMIE-PCA configuration with $conf_{pca}$ and AMIE-Conf with $conf$ measures respectively. For a fair comparison, we set the two configurations of AMIE and our system to generate rules with at most three positive atoms in the body and filtered them based on minimum confidence of 0.1, head coverage of 0.01 and rule support of 10 in case of FB15K and 2 in case of Wiki44K. We then filtered out all rules with $conf(r, \mathcal{G}) = 1$, as they do not produce any predictions.

Table 2 shows the number of facts (see the *Facts* column) predicted by the set R of *top-k* rules in the described settings and their prediction precision $pred\_prec_{OW}(R)$ (see the *Prec.* column). The size of the random sample outside $\mathcal{G}^i_{appr}$ is 20. We can observe that on FB15K, RuLES consistently outperforms both AMIE configurations. The *top-20* rules have the highest precision difference (outperforming AMIE-PCA and AMIE-Conf by 72% and 37% respectively). This is explained by the fact that the hybrid embedding quality penalizes rules with higher number of false predictions. For Wiki44K, RuLES is capable of achieving better precision in most of the cases. Notably, for the *top-20* rules RuLES predicted significantly more facts then competitors yet with a high precision.

In table 3, we compare RuLES with the recently developed NeuralLP system [43]. For this we used the Family dataset offered by the authors[3] with 28K facts over 3K entities and 12 relations. Starting from the *top-20* rules RuLES is capable of achieving significantly better precision. For the *top-10* rules the precision of NeuralLP is slightly better, but RuLES predicts many more facts.

More experiments and analysis on different datasets are provided in the technical report at `http://people.mpi-inf.mpg.de/~gadelrab/RuLES/`.

4.4 RuLES for Exception-Aware Rule Learning

In this experiment, we aim at evaluating the effectiveness of RuLES for learning exception-aware rules. First, consider in Table 4 examples of such rules learned by RuLES over Wiki44K dataset. The first rule r^1 says that a person is a citizen of the country where

[3] https://github.com/fanyangxyz/Neural-LP

top-k	FB15K				Wiki44K			
	RUMIS		**RuLES**		**RUMIS**		**RuLES**	
	$Facts$	$Prec.$	$Facts$	$Prec.$	$Facts$	$Prec.$	$Facts$	$Prec.$
20	672	0.95	34	0.97	5844	0.93	5640	0.93
50	1797	0.94	158	0.99	8585	0.83	13333	0.84
100	2672	0.94	434	0.99	21081	0.76	25265	0.81
200	4103	0.87	1155	0.96	50957	0.51	43677	0.67
500	13439	0.76	5466	0.90	–	–	–	–

Table 5: Prediction precision for the *top-k* rules learned by RUMIS and RuLES.

top-k	FB15K				Wiki44K			
	RUMIS		**RuLES**		**RUMIS**		**RuLES**	
	$Facts$	$Prec.$	$Facts$	$Prec.$	$Facts$	$Prec.$	$Facts$	$Prec.$
20	76	0.70	111	0.68	63	0.47	81	0.94
50	126	0.51	435	0.74	191	0.28	611	0.69
100	183	0.43	680	0.76	543	0.49	1698	0.79
200	310	0.30	1112	0.87	4861	0.40	3175	0.80
500	1155	0.53	3760	0.59	–	–	–	–

Table 6: Revision precision for the *top-k* rules learned by RUMIS and RuLES.

his alma mater is located, unless it is a research institution, since most researchers in universities are foreigners. The second rule r^2 states that the scriptwriter of some artistic work is also the scriptwriter of its sequel unless it is a TV series, which actually reflects the common practice of having several screenwriters for different seasons. Additionally, r^3 encodes that someone belonged to a noble family if his/her spouse is also from the same noble family, excluding the Chinese dynasties.

To quantify the quality of RuLES in learning non-monotonic rules, we compare the Conf-SSP configuration of RuLES (with embedding weight $\lambda = 0.3$) with RUMIS [34] as a non-monotonic rule revision system which finds exceptions by minimizing the conflicts between the induced rules. RUMIS learns rules of the form $r:\ h(X,Z) \leftarrow p(X,Y), q(Y,Z), not\ E$, where E is either $e(X,Z)$ or $type(X,t)$ with $t \in \mathcal{C}$. For a fair comparison we restricted RuLES to learn rules of the same form. We configured both systems setting the minimum rule support threshold to 10 and exception confidence for RuLES to 0.05. To enable the systems to learn rules with exceptions of the form $type(X,t)$, we enriched the KGs with *type* information from the original Freebase and Wikidata graphs.

Table 5 reports the number of predictions produced by a rule set R of *top-k* non-monotonic rules learned by both systems as well as their precision $pred\_prec_{OW}(R)$ with a sample of 20 prediction outside $\mathcal{G}^i_{appr}$. The results show that RuLES consistently outperforms RUMIS on both datasets. For Wiki44K, and $k \in \{50, 100\}$, the *top-k* rules produced by RuLES predicted more facts than those induced by the competitor achieving higher overall precision. Regarding the number of predictions, the converse holds for the FB15K KG; however, however the rules learned by RuLES are still more accurate.

To evaluate the quality of the chosen exceptions, we compare the $rev\_prec_{OW}(R)$ with a sample of 20 predictions. Observe that in Table 6, rules induced by RuLES prevented the generation of more facts than RUMIS. In all of the cases apart from *top-10* for FB15K, our system managed to remove a larger fraction of erroneous predictions. For

Wiki44K, RuLES consistently performs twice as good as RUMIS. In conclusion, the guidance from the embedding model exploited in our system gives us hints on which among the possible exception candidates likely correspond to noise.

5 Related Work

Inductive Logic Programming (ILP) addresses the problem of rule learning from data. In its probabilistic setting, given a set of probabilistic examples for grounded atoms and a target predicate p, the task is to learn rules for predicting probabilities of atoms for p [29,28,7]. Applying these techniques to our setting requires a full materialization of the probability function f, which quickly grows to sizes that ILP methods cannot handle.

A recently proposed differentiable ILP framework [9] has advantages over traditional ILP in its robustness to noise and errors in the underlying data. However, [9] requires negative examples, which in our case are hard to get due to the large KG size. Moreover, [9] is memory-expensive as authors admit, and cannot scale to the size of modern KGs.

Unsupervised relational association rule learning systems such as [15,12] induce logical rules from the data by mining frequent patterns and casting them into rules. In the context of KGs [12,5,34] such approaches address the incompleteness of KGs by exploiting sophisticated measures over the original graph, possibly enhanced with a schema [8,20] or constraints on the number of missing edges [33]. However, these methods do not tap any unstructured information like we do. Indeed, our hybrid embedding-based measure allows us to conveniently account for unstructured information implicitly via embeddings as well as making use of various graph-based rule metrics.

Exploiting embedding models for rule learning is a new research direction that has recently gained attention [43,42]. To the best of our knowledge, existing methods are purely statistics-based, i.e., they reduce the rule learning problem to algebraic operations on neural-embedding-based representations of a given KG. [42] constructs rules by modeling relation composition as multiplication or addition of two relation embeddings. The authors of [43] propose a differentiable system for learning models defined by sets of first-order rules that exploits a connection between inference and sparse matrix multiplication [6]. However, existing approaches pose strong restrictions on target rule patterns, which often prohibit learning interesting rules, e.g. non-chain-like or exception-aware ones, which we support.

Another line of work concerns enhancing embedding models with rules and constraints, e.g. [37,17,31,16,37]. While our direction is related, we pursue a different goal of leveraging the feedback from embeddings to improve the quality of the learned rules. To the best of our knowledge, this idea has not been considered in any prior work.

6 Conclusion

We presented a method for learning rules that may contain negated atoms from KGs that dynamically exploits feedback from a precomputed embedding model. Our approach is general in that any embedding model can be utilized including text-enhanced ones, which indirectly allows us to harness unstructured web sources for rule learning. We evaluated our approach with various configurations on real-world datasets and observed significant improvements over state-of-the-art rule learning systems.

An interesting future direction is to extend our work to more complex non-monotonic rules with higher-arity predicates, aggregates and existential variables or disjunctions in rule heads. This requires major extensions for incorporating embedding models while avoiding scalability problems.

References

1. Google's KG. http://www.google.co.uk/insidesearch/features/search/knowledge.html.
2. R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.
3. E. Auriol, M. Manago, and J. Guiot-Dorel. Cassiopée: Fehlerdiagnose von CFM 56-3 triebwerken für boing 737 flugzeuge. *KI*, 10(1):47–53, 1996.
4. A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of NIPS*, pages 2787–2795, 2013.
5. Y. Chen, S. Goldberg, D. Z. Wang, and S. S. Johri. Ontological pathfinding. In *SIGMOD*, 2016.
6. W. W. Cohen. Tensorlog: A differentiable deductive database. *CoRR*, abs/1605.06523, 2016.
7. D. Corapi, D. Sykes, K. Inoue, and A. Russo. Probabilistic rule lead-blp:conf/nips/yangyc17rning in nonmonotonic domains. In *CLIMA*, LNCS, 2011.
8. C. d'Amato, S. Staab, A. G. Tettamanzi, T. D. Minh, and F. Gandon. Ontology enrichment by discovering multi-relational association rules from ontological knowledge bases. In *SAC*, pages 333–338, 2016.
9. R. Evans and E. Grefenstette. Learning explanatory rules from noisy data. *J. Artif. Intell. Res.*, 61:1–64, 2018.
10. J. Fürnkranz and T. Kliegr. A brief overview of rule learning. In *RuleML*, pages 54–69, 2015.
11. M. H. Gad-Elrab, D. Stepanova, J. Urbani, and G. Weikum. Exception-enriched rule learning from knowledge graphs. In *Proceedings of ISWC*, pages 234–251, 2016.
12. L. Galarraga, C. Teflioudi, K. Hose, and F. M. Suchanek. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB J.*, 24(6):707–730, 2015.
13. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, pages 1070–1080. MIT Press, 1988.
14. X. Glorot, A. Bordes, J. Weston, and Y. Bengio. A semantic matching energy function for learning with multi-relational data. *CoRR*, abs/1301.3485, 2013.
15. B. Goethals and J. V. den Bussche. Relational association rules: Getting warmer. In *PDD*, 2002.
16. S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo. Jointly embedding knowledge graphs and logical rules. In *EMNLP*, 2016.
17. S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo. Knowledge graph embedding with iterative guidance from soft rules. *CoRR*, abs/1711.11231, 2017.
18. S. Im, Z. W. Ras, and H. Wasyluk. Action rule discovery from incomplete data. *Knowl. Inf. Syst.*, 25(1):21–33, 2010.
19. S. Kotsiantis and D. Kanellopoulos. Association rules mining: A recent overview. *GESTS Int. Trans. on CS and Eng.*, 32(1):71–82, 2006.
20. F. A. Lisi. Inductive Logic Programming in Databases: From Datalog to DL+log. *TPLP*, 10(3):331–359, 2010.
21. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. In *KDD-95*, 1995.

22. T. D. Minh, C. d'Amato, B. T. Nguyen, and A. G. B. Tettamanzi. Comparing rule evaluation metrics for the evolutionary discovery of multi-relational association rules in the semantic web. In *EuroGP*, 2018.
23. T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-Ending Learning. In *AAAI*, 2015.
24. M. Nickel, L. Rosasco, and T. A. Poggio. Holographic embeddings of knowledge graphs. In *AAAI*, 2016.
25. M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, 2011.
26. H. Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508, 2017.
27. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991.
28. L. D. Raedt, A. Dries, I. Thon, G. V. den Broeck, and M. Verbeke. Inducing probabilistic relational rules from probabilistic examples. In *IJCAI*, pages 1835–1843. AAAI Press, 2015.
29. L. D. Raedt and I. Thon. Probabilistic rule learning. In *ILP*, 2010.
30. Z. W. Ras and A. Wieczorkowska. Action-rules: How to increase profit of a company. In *PKDD*, 2000.
31. P. Rastogi, A. Poliak, and B. V. Durme. Training relation embeddings under logical constraints. In *KG4IR*, 2017.
32. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *Proceedings of WWW*, pages 697–706, 2007.
33. T. P. Tanon, D. Stepanova, S. Razniewski, P. Mirza, and G. Weikum. Completeness-aware rule learning from knowledge graphs. In *ISWC*, 2017.
34. H. D. Tran, D. Stepanova, M. H. Gad-elrab, F. A. Lisi, and G. Weikum. Towards nonmonotonic relational learning from knowledge graphs. *26th ILP*, 2016.
35. D. Vrandecic and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *CACM*, 57(10):78–85, 2014.
36. Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743, 2017.
37. Q. Wang, B. Wang, and L. Guo. Knowledge base completion using embeddings and rules. In *IJCAI*, 2015.
38. Z. Wang and J. Li. RDF2Rules: Learning rules from RDF knowledge bases by mining frequent predicate cycles. *CoRR*, abs/1512.07734, 2015.
39. Z. Wang and J. Li. Text-enhanced representation learning for knowledge graph. In *IJCAI*, 2016.
40. J. Wojtusiak. Rule learning in healthcare and health services research. In *Machine Learning in Healthcare Informatics*, pages 131–145. 2014.
41. H. Xiao, M. Huang, L. Meng, and X. Zhu. SSP: semantic space projection for knowledge graph embedding with text descriptions. In *AAAI*, 2017.
42. B. Yang, W. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575, 2014.
43. F. Yang, Z. Yang, and W. W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *NIPS*, pages 2316–2325, 2017.
44. K. Zupanc and J. Davis. Estimating rule quality for knowledge base completion with the relationship between coverage assumption. In *The Web Conference 2018*, 2018.