Completeness-aware Rule Learning from Knowledge Graphs

Thomas Pellissier Tanon[a], Daria Stepanova[a], Simon Razniewski[b], Paramita Mirza[a], and Gerhard Weikum[a]

[a]Max Planck Institute of Informatics
[b]Free University of Bozen-Bolzano
{tpelliss,dstepano,paramita,weikum}@mpi-inf.mpg.de,
razniewski@inf.unibz.it

Abstract. Knowledge graphs (KGs) are huge collections of primarily encyclopedic facts. They are widely used in entity recognition, structured search, question answering, and other important tasks. Rule mining is commonly applied to discover patterns in KGs. However, unlike in traditional association rule mining, KGs provide a setting with a high degree of *incompleteness*, which may result in the wrong estimation of the quality of mined rules, leading to erroneous beliefs such as all artists have won an award, or hockey players do not have children.

In this paper we propose to use (in-)completeness meta-information to better assess the quality of rules learned from incomplete KGs. We introduce completeness-aware scoring functions for relational association rules. Moreover, we show how one can obtain (in-)completeness meta-data by learning rules about numerical patterns of KG edge counts. Experimental evaluation both on real and synthetic datasets shows that the proposed rule ranking approaches have remarkably higher accuracy than the state-of-the-art methods in uncovering missing facts.

1 Introduction

Motivation. Advances in information extraction have led to general-purpose knowledge graphs (KGs) containing billions of positive facts about the world (e.g., [3,2,1,21]). KGs are widely applied in semantic web search, question answering, web extraction and many other tasks. Unfortunately, due to their wide scope, KGs are generally incomplete. To account for the incompleteness, KGs typically adopt the Open World Assumption (OWA) under which missing facts are treated as unknown rather than false.

An important task over KGs is rule learning, which is relevant for a variety of applications ranging from knowledge graph curation (completion, error detection) [24,12,10] to data mining and semantic culturonomics. However, since such rules are learned from incomplete data, they might be erroneous and might make incorrect predictions on missing facts. E.g., $r_1 : \mathit{hasChild}(X, Y) \leftarrow \mathit{worksAt}(X, Z), \mathit{educatedAt}(Y, Z)$ could be mined from the KG in Fig. 1, stating that workers of certain institutions often have children among the people educated there, as this is frequently the case for popular scientists. While r_1 is clearly not universal and should be ranked lower than the rule $r_2 : \mathit{hasSibling}(X, Z) \leftarrow \mathit{hasFather}(X, Y), \mathit{hasChild}(Y, Z)$, standard rule measures like confidence (i.e., conditional probability of the rule's head given its body) incorrectly favor r_1 over r_2 for the given KG.

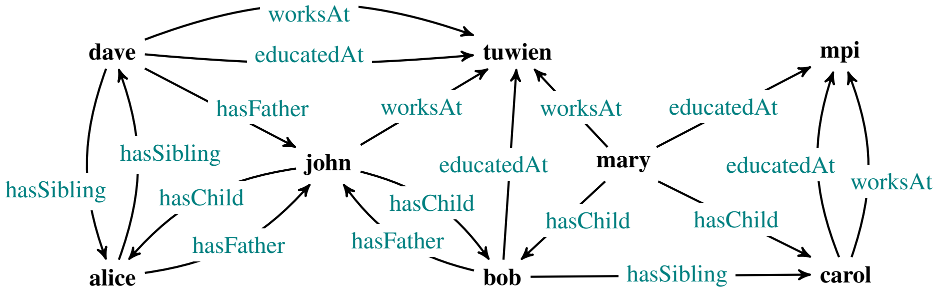

Fig. 1: Example KG

Recently, efforts have been put into detecting the concrete numbers of facts of certain types that hold in the real world (e.g., *"Einstein has 3 children"*) by exploiting Web extraction and crowd-sourcing methods [23,26]. Such meta-data provides a lot of hints about the topology of KGs, and reveals parts that should be especially targeted by rule learning methods. However, surprisingly, despite its obvious importance, to date, no systematic way of making use of such information in rule learning exists.

In this work we propose to exploit meta-data about the expected number of edges in KGs to better assess the quality of learned rules. To further facilitate this approach, we discuss a method for learning edge count information by extracting rules like *"If a person has more than 2 siblings, then his parents are likely to have more than 3 children."*

State of the art and its limitations. In [12] a completeness-aware rule scoring based on the partial completeness assumption (PCA) was introduced. The idea of PCA is that whenever at least one object for a given subject and a predicate is in a KG (e.g., *"Eduard is Einstein's child"*), then all objects for that subject-predicate pair (*Einstein's children*) are assumed to be known. This assumption was taken into account in rule scoring, and empirically it turned out to be indeed valid in real-world KGs for some topics. However, it does not universally hold, and treats cases inappropriately when edges in a graph are randomly missing. Similarly, whether to count absence of contradiction as confirmation for default rules was discussed in [8]. In [11] new completeness data was learned from a KG by taking as ground truth completeness data obtained via crowd-sourcing. The acquired statements were then used in a post-processing step of rule learning to filter out predictions that violate these statements. However, this kind of filtering does not have any impact on the quality of the mined rules and the incorrect predictions for instances about which no completeness information exists.

Contributions. This work presents the first proper investigation of how meta-information about (in-)completeness, more specifically, about the number of edges that should exist for a given subject-predicate pair in a KG, can be used to improve rule learning. The salient contributions of our work are as follows:

1. We present an approach that accounts for meta-data about the number of edges that should exist for given subject-predicate pairs in the ranking stage of rule learning.

2. We discuss a method for the automated acquisition of approximate upper and lower bounds on the number of edges that should exist in KGs.
3. We implement the proposed rule ranking measures and evaluate them both on real-world and synthetic dataset, showing that they outperform existing measures both with respect to the quality of the mined rules and the predictions they produce.[1]

2 Related Work

Rule learning. The problem of automatically learning patterns from KGs has gained a lot of attention in the recent years. Some relevant works are [12,29], which focus on learning Horn rules and either ignore completeness information, or make use of completeness by filtering out predicted facts violating completeness in a post-processing step. On the contrary we aim at injecting the statements into the learning process.

In the context of inductive and abductive logic programming, learning rules from incomplete interpretations given as a set of positive facts along with a possibly incomplete set of negative ones was studied, e.g., in [18]. In contrast to our approach, this work does not exploit knowledge about the number of missing facts, and neither do the works on terminology induction, e.g., [28]. Learning nonmonotonic rules in the presence of incompleteness was studied in hybrid settings [16,20], where a background theory or a hypothesis can be represented as a combination of an ontology and Horn or nonmonotonic rules. The main point in these works is the assumption that there might be potentially missing facts in a given dataset. However, it is not explicitly mentioned which parts of the data are (in)complete like in our setting. Moreover, the emphasis of these works is on the complex reasoning interaction between the components, while we are more concerned with techniques for deriving rules with high predictive quality from large KGs. Recent work by d'Amato et al. [4] shows how in the presence of ontologies that allow to determine incorrect facts, rules can be ranked by the ratio of correct versus incorrect predictions. In contrast to our scenario of interest, in this work, the knowledge about exact numbers of missing KG facts has not been exploited.

There are also a number of less relevant statistical approaches to completing knowledge graphs based on, e.g., low-dimensional embeddings [31] or tensor factorization [30].

Completeness information. The idea of bridging the open and closed world assumption by using completeness information was first introduced in the database world in [19,9], and later adapted to the Semantic Web in [5]. For describing such settings, the common approach is to fix the complete parts (and assume that the rest is potentially incomplete).

Recent work [11] has extended the rule mining system AMIE to mine rules about completeness, that predict in which parts a knowledge graph may be complete or incomplete. The focus of the work is on the learning of association rules like *"If someone has a date of birth but no place of birth, then the place of birth is missing."* In contrast, we reason about the missing edges trying to estimate the exact number (bounds on the number) of edges that should be present in a KG. In [11] it has also been shown that completeness information can be used to improve the accuracy of fact prediction, by pruning out in a post-processing step those facts that are predicted in parts expected to be complete. In

[1] The extended version of this paper is available as a technical report at `https://raw.githubusercontent.com/Tpt/CARL/master/technical_report.pdf`

the present paper, we take a more direct approach and inject completeness information already into the rule acquisition phase, in order to also prune away problematic rules, not only individual wrong predictions.

Our cardinality statements (e.g., John has 3 children) encode knowledge about parts of a KG that are (un)known, and thus should have points of contact with operators from epistemic logic; we leave the extended discussion on the matter for future work.

3 Preliminaries

Knowledge graphs. Knowledge graphs (KG) represent interlinked collections of factual information, and they are often encoded using the RDF data model [17]. The content of KGs is a set of $\langle \mathit{subject\ predicate\ object} \rangle$ triples, e.g., $\langle \mathit{john\ hasChild\ alice} \rangle$. For encyclopedic knowledge graphs on the semantic web, usually the open world assumption (OWA) is employed, i.e., these graphs contain only a subset of the true information.

In the following we take the unique name assumption, and for simplicity, write triples using binary predicates, like $\mathit{hasChild}(\mathit{john}, \mathit{alice})$. A signature of a KG $\mathcal{G}$ is $\Sigma_{\mathcal{G}} = \langle \mathbf{R}, \mathcal{C} \rangle$, where $\mathbf{R}$ is the set of binary predicates and $\mathcal{C}$ is the set of constants appearing in $\mathcal{G}$. Following [5], we define the gap between the available graph $\mathcal{G}^a$ and the ideal graph $\mathcal{G}^i$, which contains all correct facts over $\mathbf{R}$ and $\mathcal{C}$ that hold in the real world.

Definition 1 (Incomplete data source). *An incomplete data source is a pair $G = (\mathcal{G}^a, \mathcal{G}^i)$ of two KGs, where $\mathcal{G}^a \subseteq \mathcal{G}^i$ and $\Sigma_{\mathcal{G}^a} = \Sigma_{\mathcal{G}^i}$.*

Note that the ideal graph $\mathcal{G}^i$ is an imaginary construct whose content is generally not known. What is known instead is to which extent the available graph approximates/lacks information wrt. the ideal graph, e.g., *"Einstein is missing 2 children and Feynman none"*). We formalize this knowledge as cardinality assertions in Sec. 4.

Rule learning. Association rule learning concerns the discovery of frequent patterns in a data set and the subsequent transformation of these patterns into rules. Association rules in the relational format have been subject of intensive research in ILP (see, e.g., [7] as the seminal work in this direction) and more recently in the KG community (see [12] as the most prominent work). In the following, we adapt basic notions in relational association rule mining to our case of interest.

A *conjunctive query* Q over $\mathcal{G}$ is of the form $Q(\boldsymbol{X})$:- $p_1(\boldsymbol{X_1}), \dots, p_m(\boldsymbol{X_m})$. Its right-hand side (i.e., body) is a finite set of atomic formulas over $\Sigma_{\mathcal{G}}$, while the left-hand side (i.e., head) is a tuple of variables occurring in the body. The *answer* of Q on $\mathcal{G}$ is the set $Q(\mathcal{G}) = \{\nu(\boldsymbol{X}) \mid \nu \text{ is a function from variables to } \mathcal{C} \text{ and } \forall i : p_i(\nu(\boldsymbol{X_i})) \in \mathcal{G}\}$. As in [7], the *support* of Q in $\mathcal{G}$ is the number of distinct tuples in the answer of Q on $\mathcal{G}$.

An *association rule* is of the form $Q_1 \Rightarrow Q_2$, such that Q_1 and Q_2 are both conjunctive queries and $Q_1 \subseteq Q_2$, i.e., $Q_1(\mathcal{G}') \subseteq Q_2(\mathcal{G}')$ for any possible KG $\mathcal{G}'$. In this work we exploit association rules for reasoning purposes, and thus (with some abuse of notation) treat them as logical rules, i.e., for $Q_1 \Rightarrow Q_2$ we write $Q_2 \backslash Q_1 \leftarrow Q_1$, where $Q_2 \backslash Q_1$ refers to the set difference between Q_2 and Q_1 seen as sets of atoms.

Classical scoring of association rules is based on *rule support*, *body support* and *confidence*, which in [12] for a rule $r : \boldsymbol{H} \leftarrow \boldsymbol{B}$ with $\boldsymbol{H} = h(X, Y)$ are defined as:

$$\mathit{supp}(r) := \#(x, y) : \exists \boldsymbol{Z} : \boldsymbol{B} \wedge h(x, y) \qquad (1)$$

$$supp(\boldsymbol{B}) := \#(x,y) : \exists \boldsymbol{Z} : \boldsymbol{B} \tag{2}$$

$$conf(r) := \frac{supp(r)}{supp(\boldsymbol{B})} \tag{3}$$

where $\#\alpha : \mathcal{A}$ denotes the number of α that fulfill the condition $\mathcal{A}$, and $conf(r) \in [0,1]$. As in [12] we compute the support of the rule (body) w.r.t. to the head variables.

Example 1. Consider the KG in Fig. 1 and the rules r_1 and r_2 mined from it:

- $r_1 : hasChild(X, Y) \leftarrow worksAt(X, Z), educatedAt(Y, Z)$
- $r_2 : hasSibling(X, Z) \leftarrow hasFather(X, Y), hasChild(Y, Z)$

The body and rule supports of r_1 over the KG are $supp(\boldsymbol{B}) = 8$ and $supp(r_1) = 2$ respectively. Hence, we have $conf(r_1) = \frac{2}{8}$. Analogously, $conf(r_2) = \frac{1}{6}$. □

Support and confidence were originally developed for scoring rules over complete data. If data is missing, their interpretation is not straightforward and they can be misleading. In [12], *confidence under the Partial Completeness Assumption* (PCA) has been proposed as a measure, which guesses negative facts by assuming that data is usually added to KGs in batches, i.e., if at least one child of John is known then most probably all John's children are present in the KG. The *PCA confidence* is defined as

$$conf_{pca}(r) := \frac{supp(r)}{\#(x,y) : \exists \boldsymbol{Z} : \boldsymbol{B} \wedge \exists y' : h(x,y') \in \mathcal{G}^a} \tag{4}$$

Example 2. We obtain $conf_{pca}(r_1) = \frac{2}{4}$. Indeed, since $carol$ and $dave$ are not known to have any children in the KG, four existing body substitutions are not counted in the denominator. Meanwhile, we have $conf_{pca}(r_2) = \frac{1}{6}$, since all people that are predicted to have siblings by r_2 already have siblings in the available graph. □

Given a rule r and a KG $\mathcal{G}$ the application of r on $\mathcal{G}$ results in a rule-based graph completion defined relying on the Answer Set semantics (see [13] for details), which for positive programs coincides with the least model datalog semantics.

Definition 2 (Rule-based KG completion). *Let $\mathcal{G}$ be a KG over the signature $\Sigma_{\mathcal{G}} = \langle \mathbf{R}, \mathcal{C} \rangle$ and let r be a rule mined from $\mathcal{G}$, i.e. a rule over $\Sigma_{\mathcal{G}}$. Then the* completion of $\mathcal{G}$ *is a graph $\mathcal{G}_r$ constructed from the answer set of $r \cup \mathcal{G}$.*

Example 3. We have $\mathcal{G}^a_{r_1} = \mathcal{G} \cup \{\ hasChild(john, dave), hasChild(carol, mary),$ $hasChild(dave, dave), hasChild(carol, carol), hasChild(dave, bob),$ $hasChild(mary, dave)\ \}$. □

Note that $\mathcal{G}^i$ is the perfect completion of $\mathcal{G}^a$, i.e., it is supposed to contain all correct facts with entities and relations from $\Sigma_{\mathcal{G}^a}$ that hold in the current state of the world. The goal of rule-based KG completion is to extract from $\mathcal{G}^a$ a set of rules $\mathcal{R}$ such that $\cup_{r \in \mathcal{R}} \mathcal{G}^a_r$ is as close to $\mathcal{G}^i$ as possible.

4 Completeness-aware Rule Scoring

Scoring and ranking rules are core steps in association rule learning. A variety of measures for ranking rules have been proposed, with prominent ones being confidence, conviction and lift. The existing (in-)completeness-aware rule measure in the KG context (the PCA confidence (4)) has two apparent shortcomings: First, it only counts as counterexamples those pairs (x, y) for which at least one $h(x, y')$ is in $\mathcal{G}^a$ for some y' and a rule's head predicate h. Thus, it may incorrectly give high scores to rules predicting facts for very incomplete relations, e.g., *place of baptism*. Second, it is not suited for data in non-functional relations that is not added in batches, such as awards, where the important ones are added instantly, while others much slower or even possibly never.

Thus, in this work we focus on the improvements of rule scoring functions by making use of the extra (in-)completeness meta-data. Before dwelling into the details of our approach we discuss the formal representation of such meta-data.

Cardinality statements. Overall, one can think of 6 different cardinality templates obtained by fixing subject, predicate or object in a triple and report the number of respective facts that hold in $\mathcal{G}^i$. E.g., for $\langle john\ hasChild\ mary\rangle$ we can count (1) children of $john$; (2) edges from $john$ to $mary$; (3) incoming edges to $mary$; (4) facts with $john$ as a subject; (5) facts over $hasChild$ relation; (6) facts with $mary$ as an object.

In practice, numerical statements for templates (1) and (3) can be obtained using web extraction techniques [23], from functional properties of relations or from crowdsourcing. For other templates things get trickier; one might be able to learn them from the data or they could be defined by domain experts in topic-specific KGs. We leave this issue for future work, and focus here only on templates (1) and (3), which could be rewritten as the instances of the template (1) provided that inverse relations can be expressed in a KG. For instance, $\#s : hasChild(s, john) = \#o : hasParent(john, o)$ for the predicates $hasChild$ and $hasParent$, which are inverses of one another.

We represent the (in)completeness meta-data using cardinality statements by reporting (the numerical restriction on) the absolute number of facts over a certain relation in the ideal graph $\mathcal{G}^i$. More specifically, we define the partial function num that takes as input a predicate p and a constant s and outputs a natural number corresponding to the number of facts in $\mathcal{G}^i$ over p with s as the first argument:

$$num(p, s) := \#o : p(s, o) \in \mathcal{G}^i \tag{5}$$

Naturally, the number of missing facts for a given p and s can be obtained as

$$miss(p, s) := num(p, s) - \#o : p(s, o) \in \mathcal{G}^a \tag{6}$$

Example 4. Consider the KG in Fig. 1. and the following cardinality statements for it:

- $num(hasChild, john) = num(hasChild, mary) = 3$; $num(hasChild, alice) = 1$; $num(hasChild, carol) = num(hasChild, dave) = 0$;
- $num(hasSibling, bob) = 3$; $num(hasSibling, alice) = num(hasSibling, carol) = num(hasSibling, dave) = 2$.

We then have:

- $miss(hasChild, mary) = miss(hasChild, john) = miss(hasChild, alice) = 1$; $miss(hasChild, carol) = miss(hasChild, dave) = 0$;

– $miss(hasSibling, bob) = miss(hasSibling, carol) = 2$;
$miss(hasSibling, alice) = miss(hasSibling, dave) = 1$. □

We are now ready to define the *completeness-aware rule scoring problem*. Given a KG and a set of cardinality statements, *completeness-aware rule scoring* aims to score rules not only by their predictive power on the known KG, but also wrt. the number of wrongly predicted facts in complete areas and the number of newly predicted facts in known incomplete areas.

In the following we discuss and compare three novel approaches for completeness-aware rule scoring. These are (i) the *completeness confidence*, (ii) *completeness precision* and *recall*, and (iii) *directional metric*. Henceforth, all examples consider the KG in Fig. 1, rules from Ex. 1, and cardinality statements described in Ex. 4.

4.1 Completeness Confidence

In this work we propose to explicitly rely on incompleteness information in determining whether to consider an instance as a counterexample for a rule at hand or not.

To do that, we first define two indicators for a given rule $r : h(X, Y) \leftarrow \boldsymbol{B}$, reflecting the number of new predictions made by r in incomplete ($npi(r)$) and, respectively, complete ($npc(r)$) KG parts:

$$npi(r) := \sum_{x} min(\#y : h(x, y) \in \mathcal{G}^a_r \backslash \mathcal{G}^a, miss(h, x)) \tag{7}$$

$$npc(r) := \sum_{x} max(\#y : h(x, y) \in \mathcal{G}^a_r \backslash \mathcal{G}^a - miss(h, x), 0) \tag{8}$$

Note that summation is done exactly over those entities for which $miss$ is defined. Exploiting these additional indicators for $r : h(X, Y) \leftarrow \boldsymbol{B}$ we obtain the following *completeness-aware confidence*:

$$conf_{comp}(r) := \frac{supp(r)}{supp(\boldsymbol{B}) - npi(r)} \tag{9}$$

Example 5. Obviously, the rule r_2 should be preferred over r_1. For our novel completeness confidence, we get $conf_{comp}(r_1) = \frac{2}{6}$ and $conf_{comp}(r_2) = \frac{1}{2}$, resulting in the desired rule ordering, which is not achieved by existing measures (see Ex. 1 and 2). □

Our completeness confidence generalizes both the standard and the PCA confidence:

Proposition 1. *For every KG $\mathcal{G}$ and rule r it holds that*

(i) under the Closed World Assumption (CWA) $conf_{comp}(r) = conf(r)$;
(ii) under the Partial Completeness Assumption (PCA) $conf_{comp}(r) = conf_{pca}(r)$.

In other words, if the graph is known to be fully complete, i.e., for all $p \in \mathbf{R}$, $s \in \mathcal{C}$ we have $miss(p, s) = 0$, then $conf_{comp}$ is the same as the standard confidence. Similarly, if $miss(p, s) = 0$ for such p, s pairs that at least one fact $p(s, \_) \in \mathcal{G}^a$ exists and $miss(p, s) = +\infty$ for the rest, then $conf_{comp}$ is the same as the PCA confidence.

4.2 Completeness Precision and Recall

Further developing the idea of scoring rules based on their predictions in complete and incomplete KG parts, we propose to consider the notions of *completeness precision* and *recall*[2] for rules defined in the spirit of information retrieval. Intuitively, rules having high precision are rules that predict few facts in complete parts, while rules having high recall are rules that predict many facts in incomplete ones. Rule scoring could then be based on any weighted combination of these two metrics.

Formally, we define the precision and recall of a rule $r : h(X, Y) \leftarrow \boldsymbol{B}$ as follows:

$$precision_{comp}(r) = 1 - \frac{npc(r)}{supp(\boldsymbol{B})} \tag{10}$$

$$recall_{comp}(r) = \frac{npi(r)}{\sum_s miss(h, s)} \tag{11}$$

The *recall measure* is similar to classical support measures, but now expresses how many facts on KG parts known to be incomplete, are generated by the rule (the more the better). The *precision measure*, in turn, assesses how many of the generated facts are definitely wrong, namely those in complete parts (the more of these, the worse the rule). In fact, this is an upper bound on the precision, as the other facts cannot be evaluated.

Example 6. It holds that $npi(r_1) = 2$, $npc(r_1) = 4$, while $npi(r_2) = 4$, $npc(r_2) = 1$, resulting in $precision_{comp}(r_1) = 0.5$, $recall_{comp}(r_1) \approx 0.67$, and $precision_{comp}(r_2) \approx 0.83$, $recall_{comp}(r_2) \approx 0.67$, which lead to the expected relative rule ordering. □

Limitations. While precision and recall are insightful when there are sufficiently many predictions made in (in-)complete parts, they fail when the number of (in-)completeness statements in comparison with the KG size is small. Consider, for instance, a rule that predicts 1000 new facts over $hasChild$ relation, out of which 2 are in complete, and 2 are in incomplete parts, and overall 1 million children are missing. This would imply a precision of 99.8%, and a recall of 0.0002%, both of which are not very informative.

Therefore, next we propose to look at the difference between expected numbers of predictions in complete and incomplete parts, or simply at their ratio.

4.3 Directional Bias

If rule mining does make use of completeness information, and both do not exhibit any statistical bias, then intuitively the rule predictions and the (in)complete areas should be statistically independent. On the other hand, correlation between the two indicates that the rule-mining is *(in)completeness-aware*.

Example 7. Suppose in total a given KG stores 1 million humans, and we know that 10,000 (1%) of these are missing some children (incompleteness information), while we also know that 1000 of the persons are definitely complete for children (0.1%). Let the set of rules mined from a KG predict 50,000 new facts for the $hasChild$ relation. Assuming

[2] For brevity we skip the word ”completeness” if clear from the context.

independence between predictions and (in)completeness statements, we would expect 1% out of 50,000, i.e., 500 facts to be predicted in the incomplete areas and 0.1%, i.e., 50 in the complete KG parts. If instead we find 1000 children predicted for people that are missing correspondingly many children, and 10 for people that are not missing these, the former deviates from the expected value by a factor of 2, and the latter by a factor of 5.

Following the intuition from the above example, we propose to look at the extent of the non-independence to quantify the (in)completeness-awareness of rule mining. Let us consider predictions made by rules in a given KG, where *E(#facts)* is the expected number of predictions and $\alpha = 0..1$ is the weight given to completeness versus incompleteness. Then the directional coefficient of a rule r is defined as follows:

$$direct\_coef(r) := \alpha \cdot \frac{E(npc(r))}{npc(r)} + (1-\alpha) \cdot \frac{npi(r)}{E(npi(r))} \tag{12}$$

Unlike the other measures that range from 0 to 1, the directional coefficient takes values between 0 and infinity, where 1 is the default. If the ratio between the KG size and the size of the (in)complete parts is the same as the ratio between the predictions in the (in)complete parts and their total number, i.e., if the directional coefficient is 1, then the statements do not influence the rule at all. The higher is the *directional coefficient*, the more "*completeness-aware*" the rules are.

In practice, expected values might be difficult to compute, and statistical independence is a strong assumption. An alternative that does not require knowledge about expected values is to directly measure the proportion between predictions in complete and incomplete parts. We call this the *directional metric*, which is computed as

$$direct\_metric(r) := \frac{npi(r) - npc(r)}{2 \cdot (npi(r) + npc(r))} + 0.5 \tag{13}$$

The metric is based on the same ideas as the directional coefficient, but does not require knowledge about the expected number of predictions in complete/incomplete KG parts. It is designed to range between 0 and 1 again, thus allowing convenient weighting with other $[0,1]$ measures. The directional metric of a rule that predicts the same number of facts in incomplete as in complete parts is 0.5, a rule that predicts twice as many facts in incomplete parts has a value of 0.66, and so on.

Since the real-world KGs are often highly incomplete, it might be reasonable to put more weight on predictions in complete parts. This can be done by multiplying predictions made in complete parts by a certain factor. We propose to consider the combination of a weighted existing association rule measure, e.g., confidence or conviction and the directional metric, with the weighting factor $\beta = 0..1$. Using confidence, we obtain

$$weighted\_dm(r) = \beta \cdot conf(r) + (1-\beta) \cdot direct\_metric(r) \tag{14}$$

Example 8. We get $direct\_metric(r_1) \approx 0.33$ and $direct\_metric(r_2) = 0.8$. For $\beta = 0.5$ and confidence from Ex. 1, $weighted\_dm(r_1) \approx 0.29$ and $weighted\_dm(r_2) \approx 0.48$. □

5 Acquisition of Numerical Statements

As we have shown, exploitation of numerical (in-)completeness statements is very beneficial for rule quality assessment. A natural question is where to acquire such statements from in real-world settings. Various works have shown that numerical assertions can be frequently found on the Web [5], obtained via crowd-sourcing [6], text mining [22] or completeness rule mining [11]. We believe that mining numerical correlations concerning KG edges and then assembling them into rules is a valuable and a modular approach to obtain further completeness information, which we sketch in what follows.

We start with an available KG $\mathcal{G}^a$ and some statements of the form (5).

Step 1. For every cardinality $num(p, s) = k$, we create the facts $p_{\leq k}(s)$ and $p_{\geq k}(s)$. For the pairs $p \in \mathbf{R}, s \in \mathcal{C}$ with no available cardinality statements we construct the facts $p_{\geq \# o : p(s,o) \in \mathcal{G}^a}(s)$, encoding that outgoing p-edges from s might be missing in $\mathcal{G}^a$, as the graph is believed to be incomplete by default. Here, p_{card} with $card \in \{\leq \_, \geq \_\}$ are fresh unary predicates not present in $\Sigma_{\mathcal{G}^a}$, which describe (bounds on) the number of outgoing p-edges for a given constant. We store all constructed facts over p_{card} in $\mathcal{S}$.

We then complete the domain of each p_{card} predicate as follows. For every $p_{\leq k}(s) \in \mathcal{S}$, if $p_{\leq k'}(s') \in \mathcal{S}$ for some $s' \in \mathcal{C}$ and $k' > k$, we construct the rule $p_{\leq k'}(X) \leftarrow p_{\leq k}(X)$. Similarly, for every $p_{\geq k}(s) \in \mathcal{S}$, if $p_{\geq k'}(s') \in \mathcal{S}$ where $k' < k$, we create $p_{\geq k'}(X) \leftarrow p_{\geq k}(X)$. The constructed rules are then applied to the facts in $\mathcal{S}$ to obtain an extended set $\mathcal{G}^{card}$ of facts over p_{card}. The latter step is crucial when using a rule mining system that is not doing arithmetic inferences (like $x > 4$ implies $x > 3$).

Step 2. We then use such a standard rule learning system, AMIE [12], on $\mathcal{G}^a \cup \mathcal{G}^{card}$ to mine rules like:

(1) $p_{card}(X) \leftarrow p'_{card}(X)$
(2) $p_{card}(X) \leftarrow p'_{card}(X), p''_{card}(X)$
(3) $p_{card}(X) \leftarrow p'_{card}(X), r(X, Y)$
(4) $p_{card}(X) \leftarrow p'_{card}(X), r(X, Y), p''_{card}(Y)$
(5) $p_{card}(X) \leftarrow r(X, Y), p''_{card}(Y)$

We rank the obtained rules based on confidence and select the top ones into the set $\mathcal{R}$.

Step 3. Finally, in the last step we use the obtained ruleset $\mathcal{R}$ to derive further numerical statements together with weights assigned to them. For that we compute $\mathcal{G}' = \bigcup_{r \in \mathcal{R}} \{\mathcal{G}^{card} \cup \mathcal{G}^a\}_r$. The weights of the statements are inherited from the rules that derived them. We then employ two simple heuristics: (i) Given multiple rules predicting the same fact, the highest weight for it is kept. We then post-process predictions made by different rules for the same subject-predicate pair as follows. (ii) If $p_{\leq k}(s), p_{\geq k'}(s) \in \mathcal{G}'$ for $k' > k$, we remove from $\mathcal{G}'$ predictions with the lowest weight thus resolving the conflict on the numerical bounds.

From the obtained graph we reconstruct cardinality statements as follows.

- Given $p_{\leq k}(s), p_{\geq k}(s) \in \mathcal{G}'$ with weights w and w' we create a cardinality statement $num(p, s) = k$ with the weight $min(w, w')$.
- If $p_{\leq k}(s), p_{\geq k'}(s) \in \mathcal{G}'$ for $k' < k$, then we set $k' \leq num(p, s) \leq k$.
- Among two facts $p_{\leq k}(s), p_{\leq k'}(s)$ (resp. $p_{\geq k}(s), p_{\geq k'}(s)$) with $k < k'$ (resp. $k > k'$) the first ones are kept and represented similar to 5.

Regular facts in $\mathcal{G}'$ are similarly translated into their numerical representations.

Example 9. Consider the KG in Fig. 1 and the following cardinality statements for it: $num(\mathit{hasChild}, \mathit{john}) = num(\mathit{hasSibling}, \mathit{bob}) = 3$. Among others, $\mathcal{G}^{card}$ contains the facts: $\mathit{hasChild}_{\geq 3}(\mathit{john})$, $\mathit{hasSibling}_{\geq 3}(\mathit{bob})$, $\mathit{hasChild}_{\geq 2}(\mathit{mary})$, $\mathit{hasChild}_{\geq 2}(\mathit{john})$, $\mathit{hasSibling}_{\geq 2}(\mathit{bob})$, $\mathit{hasSibling}_{\geq 1}(\mathit{dave})$, and $\mathit{hasSibling}_{\geq 1}(\mathit{alice})$. On the graph $\mathcal{G}^a \cup \mathcal{G}^{card}$, the confidence of $\mathit{hasSibling}_{\geq 2}(X) \leftarrow \mathit{hasFather}(X, Y), \mathit{hasChild}_{\geq 3}(Y)$ is $\frac{1}{3}$ and 1 for $\mathit{hasSibling}_{\geq 1}(X) \leftarrow \mathit{hasFather}(X, Y), \mathit{hasChild}_{\geq 3}(Y)$. □

Ideally, provided that sufficiently many similar numerical correlations about edge numbers are extracted, one can induce more general hypothesis involving arithmetic functions like the number of person's siblings is bounded by the number of his parents' children plus 1 or the sum of person's brothers and sisters equals the number of his siblings. We leave these more complex generalizations for future work. Similarly, the employed heuristic provide potential for more advanced voting/weighting schemes and inconsistency resolution in the case of conflicting cardinality assertions.

6 Evaluation

6.1 Completeness-aware Rule Learning

We have implemented our **c**ompleteness-**a**ware **r**ule **l**earning approach into a C++ system prototype CARL[3], following a standard relational learning algorithm implementation such as [14]. While our general methodology can be applied to mining rules of arbitrary form, in the evaluation we focus only on rules of the form

$$r(X, Z) \leftarrow p(X, Y), q(Y, Z) \qquad (15)$$

We aim at comparing the predictive quality of the top k rules mined by our completeness-aware approach with the ones learned by standard rule learning methods: (1) AMIE [12] (PCA confidence) and (2) WarmeR [14] (standard confidence).

Dataset. We used two datasets for the evaluation: (i) *WikidataPeople*, which is a dataset we have created from the Wikidata knowledge graph, containing 2.4M facts over 9 predicates[4] about biographical information and family relationships of people; and (ii) *LUBM*, which is a synthetic dataset describing the structure of a university [15].

For the WikidataPeople dataset, the approximation of the ideal KG ($\mathcal{G}^i$) is obtained by exploiting available information about inverse relations (e.g., $\mathit{hasParent}$ is the inverse of $\mathit{hasChild}$), functional relations (e.g., $\mathit{hasFather}$, $\mathit{hasMother}$) as well as manually hand-crafted solid rules from the family domain like

$$\mathit{hasSibling}(X, Y) \leftarrow \mathit{hasParent}(X, Z), \mathit{hasParent}(Y, Z), X \neq Y.^5$$

From WikidataPeople $\mathcal{G}^i$ containing 5M facts, we acquired cardinality statements by exploiting properties of functional relations, e.g., $\mathit{hasBirthPlace}$, $\mathit{hasFather}$, $\mathit{hasMother}$ must be uniquely defined, and everybody with a $\mathit{hasDeathDate}$ has a $\mathit{hasDeathPlace}$. For the other relations, the PCA [12] is used. This resulted in 10M cardinality statements.

[3] The source code and all the data are available at `https://github.com/Tpt/CARL`.

[4] $\mathit{hasFather}$, $\mathit{hasMother}$, $\mathit{hasStepParent}$, $\mathit{hasSibling}$, $\mathit{hasSpouse}$, $\mathit{hasChild}$, $\mathit{hasBirthPlace}$, $\mathit{hasDeathPlace}$, and $\mathit{hasNationality}$

[5] see `https://github.com/Tpt/CARL/tree/master/eval/wikidata` for details

Rule r	$conf(r)$	$conf_{pca}(r)$	$conf_{comp}(r)$	$dir\_metric(r)$
$hasSibling(X,Z) \leftarrow hasSibling(X,Y),$ $hasSibling(Y,Z)$	0.10	0.10	0.89	0.98
$hasStepParent(X,Z) \leftarrow hasMother(X,Y),$ $hasSpouse(Y,Z)$	0.0015	0.48	0.0015	0.38

Table 1: Example of rules mined from WikidataPeople with global ratio of 0.5

LUBM $\mathcal{G}^i$, with 1.2M facts, was constructed by running the LUBM data generator for 10 universities, removing all `rdf:type` triples and introducing inverse predicates. 464K cardinality statements were obtained by counting the number of existing objects for each subject-predicate pair, i.e., assuming the PCA on the whole dataset.

Experimental setup. To assess the effect of our proposed measures, we first construct versions of the available KG ($\mathcal{G}^a$) by removing parts of the data from $\mathcal{G}^i$ and introducing a synthetic bias in the data (i.e., leaving many facts in $\mathcal{G}^a$ for some relations and few for others). The synthetic bias is needed to simulate our scenario of interest, where some parts of $\mathcal{G}^a$ are very incomplete while others are fairly complete, which is indeed the case in real world KGs. In Wikidata, for instance, only for 3% of non-living people sibling information is reported, while children data is known for 4%.

We proceed in two steps: First, we define a *global ratio*, which determines a uniform percentage of data retained in the available graph. To further refine this, we then factor a *predicate ratio* individually for each predicate. For the WikidataPeople KG, this ratio is chosen as *(i)* 0.8 for $hasFather$ and $hasMother$; *(ii)* 0.5 for $hasSpouse$, $hasStepParent$, $hasBirthPlace$, $hasDeathPlace$ and $hasNationality$; *(iii)* 0.2 for $hasChild$; and *(iv)* 0.1 for $hasSibling$. For the LUBM dataset, the predicate ratio is uniformly defined as 1 for regular predicates and 0.5 for inverse predicates.

For a given predicate, the final ratio of facts in $\mathcal{G}^a$ retained from those in $\mathcal{G}^i$ is then computed as $min(1, 2 * k * n)$, where k is the predicate ratio and n is the global ratio.

The assessment of the rules learned from different versions of the available KG is performed by comparing rule predictions with the approximation of $\mathcal{G}^i$. More specifically, every learned rule is assigned a *quality score*, defined as the ratio of the number of predictions made by the rule in $\mathcal{G}^i \setminus \mathcal{G}^a$ over the number of all predictions outside $\mathcal{G}^a$.

$$quality\_score(r) = \frac{|\mathcal{G}^a_r \cap \mathcal{G}^i \setminus \mathcal{G}^a|}{|\mathcal{G}^a_r \setminus \mathcal{G}^a|} \tag{16}$$

This scoring naturally allows us to control the percentage of rule predictions that hit our approximation of $\mathcal{G}^i$, similar to standard recall estimation in machine learning.

Results. From every version of the available KG we have mined rules of the form (15) and kept only rules r with $conf(r) \geq 0.001$ and $supp(r) \geq 10$, whose *head coverage*[6] is greater than 0.001. Figure 2 shows the number of kept rules and their average support (1) for each global ratio used for generating $\mathcal{G}^a$.

Evaluation results for WikidataPeople and LUBM datasets are in Figure 3. The horizontal axis displays the global ratio used for generating $\mathcal{G}^a$. We compared different rule ranking methods as previously discussed, including standard confidence (3), PCA

[6] *Head coverage* is the ratio of the number of predicted facts that are in $\mathcal{G}^a$ over the number of facts matching the rule head.

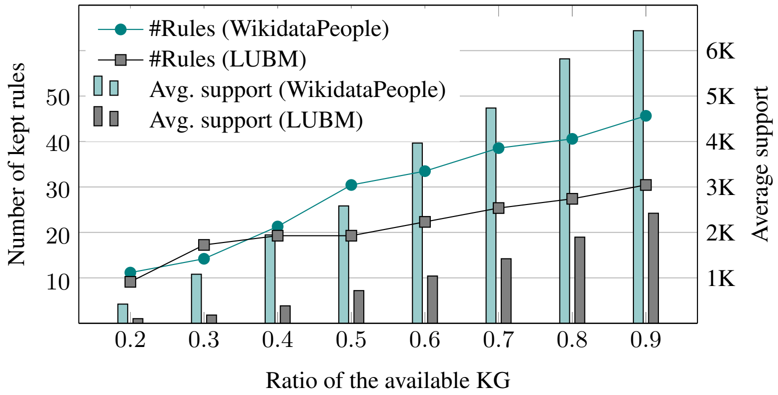

Fig. 2: Number of kept rules (#Rules) and their average support for WikidataPeople and LUBM datasets

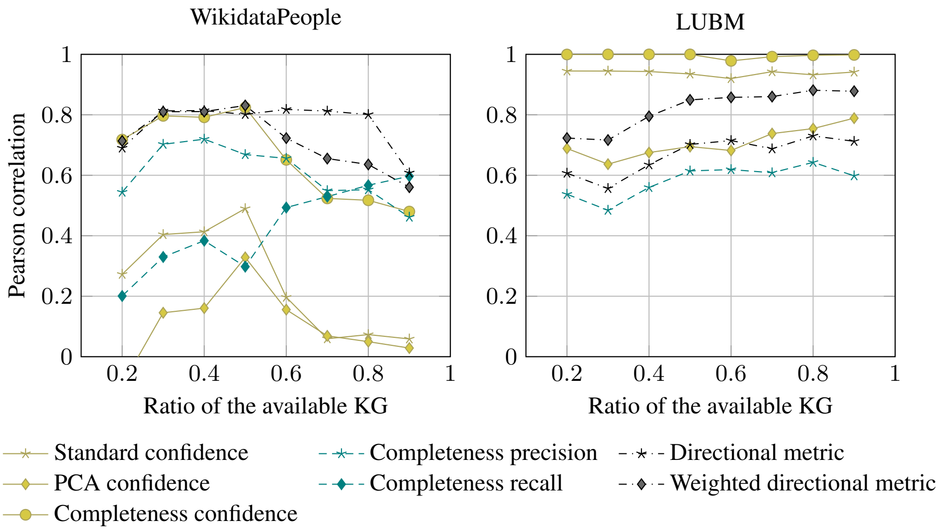

Fig. 3: Evaluation results for WikidataPeople and LUBM datasets

confidence (4), completeness confidence (9), completeness precision (10), completeness recall (11), directional metric (13) and weighted directional metric ($\beta = 0.5$) (14). The Pearson correlation factor[7] (vertical axis) between each ranking measure and the rules quality score (16) is used to evaluate the measures' effectiveness. We measured the Pearson correlation, as apart from the ranking order (captured by, e.g., the Spearman's rank correlation), the absolute values of the measures are also insightful for our setting.

[7] The Pearson correlation factor between two variables X and Y is defined by $\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$ with cov being the covariance and σ the standard deviation.

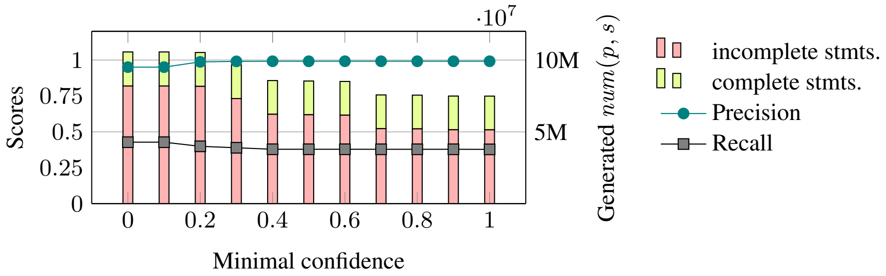

Fig. 4: Number of (in-)complete statements for generated cardinalities $num(p, s)$, and quality of predicted cardinalities.

Since facts are randomly missing in the considered versions of $\mathcal{G}^a$, the PCA confidence performs worse than the standard confidence for given datasets, while our completeness confidence significantly outperforms both (see Tab. 1 for examples).

For the WikidataPeople KG, directional metric, weighted directional metric and completeness confidence show the best results, followed by completeness precision. For the LUBM KG, the completeness confidence outperforms the rest of the measures, followed by the standard confidence and the weighted directional metric. Correlation for completeness recall in the LUBM dataset behaved erratic and was slightly negative, thus is not displayed at all. We conjecture that completeness recall might be unsuited in certain settings, because it may reward rules that predict many facts, irrespective of whether these facts are true or false. It is noteworthy that the standard confidence performs considerably better on the LUBM KG with correlation factor higher than 0.9 than on the WikidataPeople KG. Still, completeness confidence shows better results, reaching a nearly perfect correlation of 0.99. We hypothesize that this is due to the bias between the different predicates of the LUBM KG being less strong than in the WikidataPeople KG, where some predicates are missing a lot of facts, while others just a few. Completeness precision, directional metric and weighted directional metric outperform PCA confidence for most settings on the WikidataPeople KG.

6.2 Automated Acquisition of Cardinality Statements

To evaluate our method for automated acquisition of cardinality statements from a KG we reused the WikidataPeople dataset–without completing the data.

Dataset. We have collected around 282K cardinality statements from various sources:

- Wikidata schema, i.e., $hasFather$, $hasMother$, $hasBirthPlace$, and $hasDeathPlace$ are functional properties and, thus, should have at most one value.
- The 7.5K values of the Wikidata predicate $numberOfChildren$;
- 663 $novalue$ statements from Wikidata;
- 86K cardinality statements from [23] for the *hasChild* predicate of Wikidata;
- 182K cardinality statements are extracted from human-curated and complete Freebase facts (1.6M). The mapping to Wikidata has been done using tools from [25].

Experimental Setup. We set aside random 20% of the cardinality statements as validation set, while the rest were incorporated into the WikidataPeople KG, as explained in Sec. 5. We then ran our rule learning algorithm to mine cardinality rules. Rules with support less than 200 or confidence smaller than 0.01 were pruned out. Examples of mined rules along with their standard confidences include

- $hasSibling_{\geq 3}(x) \leftarrow hasSibling(x,y), hasSibling_{\geq 4}(y)$: 0.97
- $hasChild_{\geq 3}(x) \leftarrow hasFather(y,x), hasSibling(y)_{\geq 4}(y)$: 0.90.

The learned rules were then applied to the enriched WikidataPeople KG to retrieve new exact cardinalities $num(p,s)$ by only keeping (p,s) pairs where the higher and lower bounds matched. The minimum of the standard confidence of the best rules used to get the upper and lower bounds were assigned as the final confidence of each $num(p,s)$.

Results. We aim to evaluate whether we can accurately recover the cardinality statements in the validation set–as the gold standard–by utilizing the learned cardinality rules. For different minimal confidence thresholds, the quality of the predicted cardinalities is measured with standard precision and recall, which is presented in Figure 4. We get a nearly perfect precision and a fair recall (around 40%) for the generated cardinalities, which amount to 7.5M-10M depending on the threshold. Around one third of $num(p,s)$ statements indicate completeness of the KG for given (p,s) pairs. If we remove the schema information from the KG, we get lower precision (around 70%) and recall (around 1%) before a minimal confidence of 0.6, and similar values after.

7 Conclusion and Future Work

We have defined the problem of learning rules from incomplete KGs enriched with the exact numbers of missing edges of certain types, and proposed three novel rule ranking measures that effectively make use of the meta-knowledge about complete and incomplete KG parts: *completeness confidence, precision/recall* and the (weighted) *directional metric*. Our measures have been injected in the rule learning prototype CARL and evaluated on real-world and synthetic KGs, demonstrating significant improvements both w.r.t. the quality of mined rules and predictions they produce. Moreover, we have proposed a method for acquiring cardinality meta data about edge counts from KGs.

For future work, we plan to encode the cardinality information into background knowledge, e.g., using qualified role restrictions in OWL ontologies and exploit it to get rid of faulty rules that introduce inconsistencies. Another interesting further direction is to learn general correlations about edge counts that include mathematical functions, e.g., the number of siblings should be equal to the sum of the number of sisters and brothers.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A nucleus for a web of open data. In: ISWC. pp. 722–735 (2007)
2. Bollacker, K.D., Cook, R.P., Tufts, P.: Freebase: A Shared Database of Structured General Human Knowledge. In: AAAI. pp. 1962–1963 (2007)
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E.R.H., Mitchell, T.M.: Toward an Architecture for Never-Ending Language Learning. In: AAAI. pp. 2302-2310 (2010)
4. d'Amato, C., Staab, S., Tettamanzi, A.G., Minh, T.D., Gandon, F.: Ontology enrichment by discovering multi-relational association rules from ontological knowledge bases. In: SAC. pp. 333–338 (2016)
5. Darari, F., Nutt, W., Pirrò, G., Razniewski, S.: Completeness statements about RDF data sources and their use for query answering. In: ISWC. pp. 66–83 (2013)

6. Darari, F., Razniewski, S., Prasojo, R.E., Nutt, W.: Enabling fine-grained RDF data completeness assessment. In: ICWE. pp. 170–187 (2016)
7. Dehaspe, L., De Raedt, L.: Mining association rules in multiple relations. In: ILP. vol. 1297, pp. 125–132 (1997)
8. Doppa, J. R., Sorower, S., NasrEsfahani, M., Orr, J. W., Dietterich, T. G., Fern, X., Tadepalli, P., Irvine, J.: Learning Rules from Incomplete Examples via Implicit Mention Models. ACML: 197-212 (2011)
9. Etzioni, O., Golden, K., Weld, D.S.: Sound and efficient closed-world reasoning for planning. AI 89(1-2), 113–148 (1997)
10. Gad-Elrab, M.H., Stepanova, D., Urbani, J., Weikum, G.: Exception-enriched rule learning from knowledge graphs. In: ISWC. pp. 234–251 (2016)
11. Galárraga, L., Razniewski, S., Amarilli, A., Suchanek, F.M.: Predicting completeness in knowledge bases. In: WSDM. pp. 375–383 (2017)
12. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast Rule Mining in Ontological knowledge bases with AMIE+. In: VLDB. vol. 24, pp. 707–730 (2015)
13. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proc. of ICLP/SLP. pp. 1070–1080 (1988)
14. Goethals, B., den Bussche, J.V.: Relational association rules: Getting WARMeR. In: Pattern Detection and Discovery, ESF Exploratory Workshop. pp. 125–139 (2002)
15. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. Web Semantics: Science, Services and Agents on the World Wide Web 3(2-3) (2011)
16. Józefowska, J., Lawrynowicz, A., Lukaszewski, T.: The role of semantics in mining frequent patterns from knowledge bases in description logics with rules. TPLP 10(3), 251–289 (2010)
17. Lassila, O., Swick, R.R.: Resource description framework (RDF) model and syntax specification (1999)
18. Law, M., Russo, A., Broda, K.: Inductive learning of answer set programs pp. 311–325 (2015)
19. Levy, A.Y.: Obtaining complete answers from incomplete databases. In: VLDB. pp. S 402–412 (1996)
20. Lisi, F.A.: Inductive logic programming in databases: from Datalog to DL+log. TPLP 10(3), 331–359 (2010)
21. Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: A knowledge base from multilingual wikipedias. In: CIDR (2015)
22. Mirza, P., Razniewski, S., Darari, F., Weikum, G.: Cardinal virtues: Extracting relation cardinalities from text. ACL (2017)
23. Mirza, P., Razniewski, S., Nutt, W.: Expanding Wikidata's parenthood information by 178%, or how to mine relation cardinality information. In: ISWC 2016 Posters & Demos (2016)
24. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web 8(3), 489–508 (2017)
25. Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T., Pintscher, L.: From Freebase to Wikidata: The great migration. In: Proc. of WWW. pp. 1419–1428 (2016)
26. Prasojo, R.E., Darari, F., Razniewski, S., Nutt, W.: Managing and consuming completeness information for wikidata using COOL-WD. In: COLD@ISWC (2016)
27. Razniewski, S., Suchanek, F.M., Nutt, W.: But what do we actually know. pp. 40–44 (2016)
28. Sazonau, V., Sattler, U., Brown, G.: General Terminology Induction in OWL. ISWC, 533-550 (2015)
29. Wang, Z., Li, J.: RDF2Rules: Learning rules from RDF knowledge bases by mining frequent predicate cycles. CoRR abs/1512.07734 (2015)
30. Nickel, M., Tresp, V., Kriegel, H.P. Factorizing YAGO: scalable machine learning for linked data. In WWW. pp. 271–280 (2012).
31. Wang, Zhen, et al. Knowledge Graph Embedding by Translating on Hyperplanes. AAAI, pp. 1112–1119 (2014).