MiniAMIE: Quick and Dirty Rule Mining on Knowledge Bases

Luis Galárraga
Inria, Univ. Rennes, CNRS, IRISA
UMR 6074, F-35000 Rennes, France
luis.galarraga@inria.fr

Julianne Guerbette
Univ. Rennes, CNRS, Inria, IRISA
UMR 6074, F-35000 Rennes, France
julianne.guerbette@irisa.fr

Isseïnie Sinouvassane
Univ. Rennes, CNRS, Inria, IRISA
UMR 6074, F-35000 Rennes, France
isseinie.sinouvassane@irisa.fr

Paul Viallard
Inria, Univ. Rennes, CNRS, IRISA
UMR 6074, F-35000 Rennes, France
paul.viallard@inria.fr

Abstract

Efficient rule mining on large modern knowledge graphs (KGs) is a major challenge due to the exponential search space. Current systems – especially those aiming for exhaustive mining – remain resource- and time-consuming. In this paper, we propose MiniAMIE, a rule mining approach based on the AMIE algorithm, which restricts AMIE's language bias and estimates key rule metrics using fast approximations. Our experiments on several KGs illustrate the trade-offs of this design and show that MiniAMIE achieves a substantial speed-up while maintaining some good-quality rules.

CCS Concepts

• **Computing methodologies** → *Logic programming and answer set programming*.

Keywords

Rule mining, Knowledge graphs, Link prediction

ACM Reference Format:

Luis Galárraga, Julianne Guerbette, Isseïnie Sinouvassane, and Paul Viallard. 2025. MiniAMIE: Quick and Dirty Rule Mining on Knowledge Bases. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Rule mining on knowledge graphs (KGs) is crucial in tasks such as knowledge inference, data completion, and KG compression, as well as in AI explainability. It is also computationally challenging due to the size of current KGs. This observation has motivated several research endeavors to extract logical rules, usually Horn rules, from very large KGs [1–4, 6] containing millions of entities and billions of facts. While this is good news for data providers and consumers, modern rule mining systems still require powerful computing resources, not available to everyone. The reasons are threefold. First, rule mining entails the exploration of an exponentially large space of candidate rules. Second, for each of those rules, mining algorithms must execute expensive queries to estimate rule quality. Third, speed-up usually relies on heavy in-memory indexing. Anytime algorithms [4] provide a solution to the first challenge as they allow exploration within a time budget, trading efficiency for exhaustivity. Another solution is to restrict the rule language bias so that fewer candidate rules are explored [4]. Sampling and approximative approaches for rule quality metrics can mitigate the second challenge at the expense of metric accuracy and predictive power, whereas the third challenge can be alleviated by relying on either (hybrid) in-disk storage solutions or more lightweight in-memory indexing, the former strategy known to be in tension with runtime [3, 6]. This article explores different approximative techniques applied to a well-known exhaustive rule mining algorithm: AMIE3 [3]. Our lightweight version of AMIE, called mini-AMIE restricts AMIE's language bias to closed paths (first challenge), proposes a fast approximation, based on independence assumptions, for the support and confidence of rules (second challenge), and drops some of AMIE's indexes as a consequence of the previous simplifications (third challenge). Our evaluation aims to answer three *research questions*: (i) what are the runtime and memory gains of restricting the language bias to closed paths and approximating rule metrics on KGs?; (ii) how well can rule support and confidence be approximated by the means of join counts and independence assumptions?; (iii) how much link prediction performance do we lose by using the studied techniques?

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

2 Related Work

Extracting rules from KGs is the central goal of Inductive Logic Programming (ILP) [5]. However, early ILP systems are unsuitable for modern KGs because of their limited scalability and incompatibility with the open world assumption. AMIE [2] was the first rule mining approach suitable for large and potentially incomplete KGs. AMIE is an exhaustive top-down algorithm focused on closed Horn rules. Its successors [1, 3] proposed a handful of novel heuristics to speed-up search, including fast confidence approximations to prune bad rules in advance and a lazy evaluation routine for confidence. RudiK [6] extends AMIE's language bias to include rules with negated atoms (useful for data correction) but drops out the exhaustivity constraint – like in traditional ILP. AnyBURL [4] on the other hand, is an anytime bottom-up algorithm that mines closed path rules from samples of subgraphs within a time budget.

3 Preliminaries

Knowledge Graphs. A *knowledge graph* (KG) is a set $\mathcal{K}$ of *assertions* or *facts* $\langle S, P, O \rangle \in \mathcal{E} \times \mathcal{P} \times \mathcal{E}$, also denoted as $P(S, O)$, where $\mathcal{E}$ is a set of entities and $\mathcal{P}$ is a set of binary relations or predicates. Thus, KGs model statements, such as $capital(Germany, Berlin)$, and can also be seen as labeled directed graphs that connect node entities via relation edges. Given a relation $r \in \mathcal{P}$, we define its inverse $r^{-1} = \{(O, S) : r(S, O) \in \mathcal{K}\}$.

Atoms and Rules. An atom is a fact where at least one of the arguments is replaced by a variable $v \in \mathcal{V}$, where $\mathcal{V} \cap (\mathcal{E} \cup \mathcal{P}) = \emptyset$. For instance, $B = livesIn(x, y)$ is an atom while x and y are variables. By convention, we write for S and O in a fact $P(S, O)$, variables in lowercase and entities (constants) in uppercase. An instantiation $\iota : \mathcal{V} \to \mathcal{E}$ is a partial mapping from variables to constants in a KG. Applying an instantiation ι to an atom B – denoted by $\iota(B)$ – replaces the variables in B with their associated constants in the mapping resulting in another atom or in a fact (then called a *grounded atom*). If an atom has a single constant argument like in $speaks(x, Danish)$, we say it is an *instantiated atom*. Instantiations can be naturally extended to arbitrary logical formulae on atoms. A Horn rule is an expression of the form $R : \mathbf{B} \Rightarrow H$ where the body $\mathbf{B} = \bigwedge_{1 \leq i \leq n} B_i$ is a conjunction of atoms and the head H is a single atom. A rule is safe if the head variables also appear in the rule's body. Safe rules allow for *concrete predictions*. In this work we focus on *closed rules*, i.e., safe rules where each variable appears at least in two different atoms. When each variable appears exactly in two atoms we say the rule is a *closed path rule* as there exists a unique path connecting every pair of variables in the rule as in $companySeat(x, z) \wedge cityIn(z, y) \Rightarrow companyCountry(x, y)$.

Predictions and Metrics. Rules convey regularities that hold in a KG and that can be used to extract axiomatic knowledge or to infer new assertions. These tasks, however, require measuring the extent to which that regularity holds in the KG. Given a mapping ι and a rule $R : \mathbf{B} \Rightarrow H$, we say that ι is *a match* of R iff $\iota(B_i) \in \mathcal{K}$ for every $i \in \{1, \ldots, n\}$, which we denote by $\iota \sim \mathbf{B}$. If $\iota(H) \in \mathcal{K}$ we say $\iota(H)$ is an *observed prediction* of R and ι is a *full match* of R – written $\iota \sim R$ or equivalently $\iota \sim \mathbf{B} \wedge H$. The support of a rule R in a KG $\mathcal{K}$ is the number of unique observed predictions of R, defined as:

$$supp(\mathbf{B} \Rightarrow H) = |\iota_{|v(H)} : \iota \sim (\mathbf{B} \wedge H)| = |\iota_{|H} \sim (\mathbf{B} \wedge H)|.$$

The expression $\iota_{|v(H)}$, simplified $\iota_{|H}$, denotes the projection of the mapping ι on $v(H)$, the variables in the rule's head. The support is a measure of significance. A rule with very few observed predictions will very likely constitute noise, but many observed predictions do not still guarantee that the rule is reliable to make inferences. That is the goal of the confidence metric:

$$conf(\mathbf{B} \Rightarrow H) = \frac{supp(\mathbf{B} \Rightarrow H)}{supp(\mathbf{B} \Rightarrow H) + |(\iota_{|H} \sim \mathbf{B}) \wedge \neg\eta(\iota(H))|}.$$

Here, η is a Boolean function that returns true when a fact *is not known* to be false. Rule miners make different assumptions about η, leading to different confidence scores. AMIE, for instance, uses the PCA (Partial Completeness Assumption) confidence [2].

4 MiniAMIE: Quick and Dirty Rule Mining

We now elaborate on the MiniAMIE approach. But, we first introduce the original AMIE algorithm.

4.1 The AMIE Algorithm

AMIE [2] is a top-down exhaustive rule mining approach designed to find closed Horn rules in large KGs under the open world assumption. AMIE seeds the search with general rules of the form $\top \Rightarrow r(x, y)$, which are then iteratively refined in a breadth-first-search manner by means of three mining operators:

Add dangling atom, $O_D(R)$. It returns duplicates of R with an extra atom that has a common variable with R and a fresh variable.
Add closing atom, $O_C(R)$. It returns duplicates of R with a new atom that shares both variables with R.
Add instantiated atom, $O_I(R)$. It returns duplicates of R with a new instantiated atom that shares a variable with R.

Algorithm 1 describes the AMIE algorithm. The algorithm uses a customized in-memory database with many indexes on facts (SPO, SOP, PSO, POS, OSP, OPS) and count aggregates per triple component (S, P, O) to optimize support and confidence calculations for rules. For more details about the multiple optimizations proposed to the base routine we refer the reader to [1–3].

Algorithm 1 AMIE

Require: a KB: $\mathcal{K}$, support, confidence and length thresholds: θ, γ, l
Ensure: set of closed Horn rules: $\mathscr{R}$

$q \leftarrow [\top \Rightarrow r_1(x, y), \top \Rightarrow r_2(x, y), \ldots]; \quad \mathcal{R} \leftarrow \emptyset$
while $q \neq \emptyset$ **do**
  $R \leftarrow q$.dequeue()
  if closed$(R) \wedge$ *pca-conf*$(R) \geq \lambda$ **then**
    $\mathscr{R} \leftarrow \mathcal{R} \cup \{R\}$
  end if
  for all $R_c \in (O_D \cup O_C \cup O_I)(R) \wedge |R_c| < l \wedge supp(R_c) \geq \theta$ **do**
    q.enqueue(R_c)
  end for
end while
return $\mathscr{R}$

4.2 MiniAMIE

MiniAMIE[1] implements two important simplifications to Alg. 1 to speed-up rule mining.

Restricting the language bias. We restrict the search to closed path rules as they are less numerous than closed rules and are known to retain good predictive power [4]. These are rules of the form:

$$r_n(x, x_n) \wedge \left(\bigwedge_{n \geq i > 1} r_i(x_i, x_{i-1}) \right) \wedge r_1(x_1, y) \Rightarrow r_h(x, y).$$

AnyBurl [4] focuses on closed-path rules but also mines non-closed (but still safe) rules with instantiated atoms, e.g., $wonPrize(y, z) \wedge actedIn(x, y) \Rightarrow wonPrize(x, Emmy)$. We exclude those so that our language bias lies at the intersection of AMIE's and AnyBurl's.

[1]Code & experimental data at https://github.com/dig-team/AMIE/tree/mini-amie

Approximating support and confidence. The support and confidence calculations in lines 4 and 7 of Alg. 1 may lead to expensive count queries. We therefore devise a fast approximation of those metrics for closed path rules. We first observe that rule support can be written as follows:

$$supp(\mathbf{B} \Rightarrow H) = |\iota_{|H} \sim H| \times P(\iota \sim \mathbf{B}|\iota_{|H} \sim H). \quad (1)$$

The first term is the number of unique matches of the rule's head atom, whereas the second term is the *head coverage*, the probability that an instantiation matches the rule's body given that it matches the head. Eq. (1) illustrates why support calculation can be expensive: while the first term can be computed efficiently with the indexes, we still have to check the existence of a body match for each head match. We propose to estimate head coverage using AMIE's pre-computed data statistics. We first observe that

$$P(\iota \sim \mathbf{B}|\iota_{|H} \sim H) = P(\iota \sim r_n(x, x_n) \wedge (\bigwedge_{n \geq i > 1} \iota \sim r_i(x_i, x_{i-1})) \wedge \iota \sim r_1(x_1, y) \mid \iota \sim r_h(x, y)). \quad (2)$$

If we consider each partial match $\iota \sim r(x, \cdot)$ as a random variable, the joint distribution of those matches forms an undirected graph. To estimate this joint probability efficiently, we assume that the distribution follows the structure of a Markov Random Field (MRF), which allows constant-time estimation under certain independence assumptions. We thus assume that a match $\iota \sim r_i(x, \cdot)$ in the rule is independent of all non-consecutive matches given the match of its neighbor $r_{i-1}(\cdot, x')$. That way we can factorize Eq. (2) as:

$$P(\iota \sim \mathbf{B}|\iota \sim H) = P(\iota \sim r_n(\mathbf{x}, x_n')|\iota \sim r_h(\mathbf{x}, y')) \times \prod_{n \geq i > 1} P(\iota \sim r_i(x_{i+1}', \mathbf{x_i}) \wedge \iota \sim r_{i-1}(\mathbf{x_i}, x_{i-1}')) \times P(\iota \sim r_1(x_1', \mathbf{y})|\iota \sim r_h(x', \mathbf{y})). \quad (3)$$

for some existentially quantified fresh variables $x', y', x_{n+1}', \ldots x_1'$. If we now define the following terms for two relations r and r' – using AMIE's join statistics [1] with $dom(r) = \{s : \exists o : r(s, o) \in \mathcal{K}\}$:

$$\hat{\sigma}(r, r') = \frac{|dom(r) \cap dom(r')|}{|dom(r')|} = P(\iota \sim r(x, \cdot) \mid \iota \sim r'(x, \cdot))$$

$$\sigma(r, r') = \frac{|dom(r) \cap dom(r')|}{|dom(r) \cup dom(r')|} = P(\iota \sim r(x, \cdot) \wedge \iota \sim r'(x, \cdot)),$$

we can estimate the different probabilities in Equation 3 as follows:

$$P(\iota \sim r_n(x, x_n') \mid \iota \sim r_h(x, y')) = \hat{\sigma}(r_n, r_h), \quad (4)$$

$$P(\iota \sim r_i(x_{i+1}', \mathbf{x_i}) \wedge \iota \sim r_{i-1}(\mathbf{x_i}, x_{i-1}')) = \sigma(r_i^{-1}, r_{i-1}), \quad (5)$$

$$P(\iota \sim r_1(x_1, y) \mid \iota \sim r_h(x', y)) = \hat{\sigma}(r_1^{-1}, r_h^{-1}), \quad (6)$$

Recall that in AMIE, the dangling atom operator O_D produces non-closed rules. The estimated support for these intermediate rules resembles Eq. (3) but without the first term. This is the case because the body joins the head atom only on variable y.

Finally, the PCA confidence of a rule can be written as follows [2]:

$$pca\text{-}conf(R : \mathbf{B} \Rightarrow H) = P(\iota_{|H} \sim H|\iota \sim \mathbf{B} \wedge H') \quad (7)$$

where $H' = r_h(x, y')$. In a similar vibe we approximate the PCA confidence of rules by:

$$P(\iota \sim r_h(x, y') \mid \iota \sim r_n(x, x_n')) = \hat{\sigma}(r_h, r_n) \quad (8)$$

Dataset	Density	Method	Hits@10	MRR
yago3-10	< 0.001	AnyBurl	**0.639**	**0.545**
		AMIE	0.553	0.428
		MiniAMIE (full)	0.454 (-0.099)	0.421 (-0.007)
		MiniAMIE (o.l.)	0.548 (-0.005)	0.468 (+0.040)
fb15k237	< 0.001	AnyBurl	**0.509**	0.320
		AMIE	0.499	**0.321**
		MiniAMIE (full)	0.313 (-0.186)	0.190 (-0.131)
		MiniAMIE (o.l.)	0.489 (-0.010)	0.314 (-0.007)
oblc[1]	< 0.001	AnyBurl	0.542	**0.274**
		AMIE[2]	0.393	0.188
		MiniAMIE (full)	0.206 (-0.187)	0.108 (-0.08)
		MiniAMIE (o.l.)	**0.554** (+0.161)	0.272 (+0.084)
kinship	0.064	AnyBurl	0.891	0.505
		AMIE	**0.918**	**0.550**
		MiniAMIE (full)	0.149 (-0.769)	0.071 (-0.479)
		MiniAMIE (o.l.)	0.916 (-0.002)	0.544 (-0.006)
nations	0.318	AnyBurl	**0.990**	0.499
		AMIE	0.919	0.548
		MiniAMIE (full)	0.938 (+0.019)	0.407 (-0.141)
		MiniAMIE (o.l.)	0.970 (+0.051)	**0.597** (+0.049)
wn18rr	< 0.001	AnyBurl	**0.641**	**0.572**
		AMIE	0.485	0.460
		MiniAMIE (full)	0.080 (-0.405)	0.058 (-0.402)
		MiniAMIE (o.l.)	0.488 (+0.003)	0.463 (+0.003)

Table 1: Link prediction performance of AnyBurl, AMIE and MiniAMIE. All systems were run with $\theta = 5$ and the O_I operator enabled. Best values per dataset are shown in bold. MiniAMIE performance's difference w.r.t. AMIE is shown in parentheses.

5 Evaluation

Our evaluation is designed to answer our three research questions. All experiments were run in a server equipped with a Intel Xeon Gold 5220 CPU, 96GB DDR4 RAM and a 30TB HDD.

Evaluation on Resource Consumption. To answer **RQ1**, i.e., what are the runtime and memory gains of our speed-up techniques, we ran AMIE and MiniAMIE with 4 parameter configurations on different public KGs. The configurations comprise AMIE's default settings (head coverage threshold of 0.01, $l = 3$, O_I operator off), the activation of the expensive operator O_I as well as different thresholds on support (θ) and rule length (l). Table 2 compares the systems in terms of runtime and peek memory. We also include AnyBURL, an anytime method (default time 1 min.) whose language bias is comparable to the $D + \theta + O_I$ configuration but with $l = 4$. For small datasets, e.g., wn18rr, MiniAMIE can still be faster but AMIE remains a strong baseline. In contrast, runtime gains can reach up to 5 orders of magnitude for large datasets, specially for very exhaustive settings (e.g., $l = 4$). Likewise, peek memory can be reduced up to 79% (oblc), which makes MiniAMIE apt for running in personal computers on KGs with tens of millions of facts. This is mainly because AMIE's in-memory indexes OSP, OPS, and SOP

[1] https://openbiolink.github.io/, evaluated on 1000 triples due to the high no. of rules.
[2] Using the rules extracted after 44 hours of execution.

Dataset	Method	Runtime				Memory		Precision	Pearson	
		Default (D)	D + l = 4	D+θ={500, 100, 5}	D + θ + O_I	Default	D + θ + O_I	D + θ	Supp.	Conf.
	AMIE	7.4m	>1d	10.4m	>3d	25.50 GB	24.63 GB			
caligraph	MiniAMIE	4.4m	9.8m	3.1m	2.8d	14.71 GB	14.26 GB	0.12	0.53	0.38
	AnyBurl			1 minute		-	21.28 GB			
	AMIE	2.9m	>1d	3.3m	>1d	23.21 GB	24.53 GB			
wikidata2014	MiniAMIE	2.5m	3.1m	1.5m	18.9h	14.64 GB	14.73 GB	0.11	0.19	0.60
	AnyBurl			1 minute		-	20.11 GB			
	AMIE	2.3m	2.0d	2.3m	5.7h	7.94 GB	15.82 GB			
oblc	mini-AMIE	1.1s	2.1s	1.0s	37.5m	2.16 GB	3.38 GB	0.38	0.54	0.22
	AnyBurl			1 minute		-	15.06 GB			
	AMIE	11.2s	1.0h	11.2s	33.3m	4.50 GB	4.66 GB			
yago3-10	mini-AMIE	1.1s	1.6s	0.9s	3.3m	1.09 GB	2.22 GB	0.18	0.99	0.61
	AnyBurl			1 minute		-	3.43 GB			
	AMIE	7.0s	19.7m	6.3s	2.1m	4.45 GB	4.88 GB			
fb15k237	mini-AMIE	9.4s	2.7m	4.4s	3.9m	2.34 GB	2.30 GB	0.25	0.76	0.07
	AnyBurl			1 minute		-	4.32 GB			
	AMIE	2.0s	8.7s	2.5s	8.0s	2.52 GB	4.28 GB			
wn18rr	mini-AMIE	0.3s	0.3s	0.2s	17.0s	241.5 MB	1.62 GB	0.04	-0.02	0.99†
	AnyBurl			1 minute		-	1.95 GB			
	AMIE	2.2s	53.8m	2.7s	2.1h	2.15 GB	3.66 GB			
kinship	mini-AMIE	1.4s	57.4s	1.4s	17.6s	652.2 MB	2.49 GB	0.84	0.32	0.40
	AnyBurl			1 minute		-	15.05 GB			
	AMIE	19.8s	>3h	5.4m	9.9h	2.17 GB	15.04 GB			
nations	mini-AMIE	<0.1s*	<0.1s*	6.6s	15s	77.3 MB	4.18 GB	0.62	0.78	0.51
	AnyBurl			1 minute		-	4.06 GB			

Table 2: Runtime and peak memory usage per configuration and method. No. of triples is color-coded in the dataset name into 4 groups, i.e., $> 10M$, $(1M, 10M]$, $(100k, 1M]$, $< 100k$. The first two KGs use $\theta = 500$, the last three ones $\theta = 5$; $(^*)$ means no rules were found and (†) means the result is not significant due to insufficient data.

can now be dropped thanks to our approximated metrics.

Approximation Evaluation (RQ2). The third right-most column of Table 2 reports MiniAMIE's precision at retrieving AMIE's rules for the configuration $D + \theta$. Precision is overall low, which means that MiniAMIE finds many false positive rules. That said, the correlation between the approximated and the real support (calculated for $D + \theta + O_I$ and reported on the second right-most column) for MiniAMIE's rules is often significant. The results are similar for approximated confidence among the rules also found by AMIE. This means that while MiniAMIE's approximated scores are not accurate per se, they can still serve for ranking rule predictions by likelihood.

Link Prediction Evaluation. To answer **RQ3** about the impact of lightweight rule mining in link prediction, we tested the rules mined by AMIE, MiniAMIE and AnyBurl on different link prediction benchmark datasets and report MRR and hits@10 performance in Table 1. To guarantee comparable results, we ran all systems with a minimum support threshold of 5 examples (AnyBurl's default setting) and enabled instantiated atoms in both AMIE and MiniAMIE. For the latter we consider two variants: (i) the *only language* variant (abbreviated o.l.) that mines only closed paths but still computes exact scores, and (ii) the full MiniAMIE with approximate support and confidence. We observe that mining only closed paths incurs little performance penalty in prediction performance, and can even be beneficial. This suggests that in some datasets closed non-path rules might be noisier. MiniAMIE full incurs a higher performance penalty, which varies greatly across datasets. Its approximated metrics provide even a slight performance boost for hits@10 for the dataset nations. The edge density of the nations graph (0.318) makes our independence assumption reasonable. High density, however, does not guarantee this is always the case (e.g., kinship).

6 Discussion and Conclusion

We have presented MiniAMIE, a quick and dirty version of AMIE based on fast approximations for rule quality scores – and available in AMIE's latest version. While MiniAMIE is not meant to replace SOTA systems, the proposed techniques allow for quick exploratory analysis on large KGs in small servers. Furthermore, our approximation techniques could be integrated into other mining architectures or pipelines to, for instance, prune unpromising regions of the search space or as cardinality estimators for efficient query planning. We expect these results to inform the designers of rule mining systems about the trade-offs of approximative techniques.

References

[1] L. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek. Fast Rule Mining in Ontological Knowledge Bases with AMIE+. *VLDB Journal*, 24(6), 2015.

[2] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek. AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases. In *Proceedings of the 22nd International conference on World Wide Web*, 2013.

[3] J. Lajus, L. Galárraga, and F. Suchanek. Fast and Exact Rule Mining with AMIE 3. In *European Semantic Web Conference*, pages 36–52. Springer, 2020.

[4] C. Meilicke, M. W. Chekol, P. Betz, M. Fink, and H. Stuckeschmidt. Anytime Bottom-up Rule Learning for Large-scale Knowledge Graph Completion. *VLDB Journal*, 33(1), 2024.

[5] S. Muggleton. Learning from positive data. In *Inductive Logic Programming*, 1997.

[6] S. Ortona, V. Meduri, and P. Papotti. RuDiK: Rule Discovery in Knowledge Bases. *Proceedings of the VLDB Endowment*, 11:1946–1949, Oct. 2018.

Received 17 November 2025