



RDF Knowledge Base Summarization by Inducing First-Order Horn Rules

Ruoyu Wang^{1,2}, Daniel Sun^{1,3(✉)}, and Raymond Wong¹

¹ University of New South Wales, Sydney, Australia
ruoyu.wang2@unsw.edu.au, wong@cse.unsw.edu.au

² Shanghai Jiao Tong University, Shanghai, China
wang.ruoyu@sjtu.edu.cn

³ Enhitech LLC., Shanghai, China
danielwsun@gmail.com

Abstract. RDF knowledge base summarization produces a compact and faithful abstraction for entities, relations, and ontologies. The summary is critical to a wide range of knowledge-based applications, such as query answering and KB indexing. The patterns of graph structure and/or association are commonly employed to summarize and reduce the number of triples. However, knowledge coverage is low in state-of-the-art techniques due to limited expressiveness of patterns, where variables are under-explored to capture matched arguments in relations. This paper proposes a novel summarization technique based on first-order logic rules where quantified variables are extensively taken into account. We formalize this new summarization problem to illustrate how the rules are used to replace triples. The top-down rule mining is also improved to maximize the reusability of cached results. Qualitative and quantitative analyses are comprehensively done by comparing our technique against state-of-the-art tools, with showing that our approach outperforms the rivals in conciseness, completeness, and performance.

Keywords: Data summarization · RDF KB summarization · Knowledge graphs · Logic rule mining · Rule-based approaches

1 Introduction

Data summarization [1] is to extract, from the source, a subset or a compact abstraction that includes the most representative features or contents. Summarization of RDF Knowledge Bases (KBs) are also being studied for over a decade [3], especially after the concepts of semantic web and linked data are widely accepted, and the online data amount grows unexpectedly large.

To serve the purpose of concise and faithful summarization, structural methods [7, 16] are among the first attempts where techniques are borrowed from general graph mining approaches. Statistical and deep learning techniques [10, 15] are also welcome in the research to alleviate the impact of noise and capture latent correlations. However, the above methodologies cannot provide the

overview in an interpretable way and, in the meantime, be dependable in reasoning and deduction. Thus, approaches based on association patterns and logic rules are studied in more recent works [2, 14, 20].

Current pattern-based and rule-based methods summarize KGs and produce schematic views of the data. A technique for Logical Linked Data Compression (LLC) [14] has been proposed to extract association rules that represent repeated entities or relation-entity pairs at a lower cost. Labeled frequent graph structures are encoded as bit strings in KGIST [2] and summarized from the perspective of bit compression. Nevertheless, the extracted patterns fail to conclude general patterns with arbitrary variables and thus cover only a tiny part of the factual knowledge. First-order logic rules, such as Horn rules, are a promising upgrade where universally and existentially quantified variables are extensively supported, but the rules have not yet been used for the summarization purpose. First-order logic rules have been proved useful to KGs in knowledge-based applications, such as KG completion [9], and show competitive capabilities. However, the performance turns out to be the cost of expressiveness. For example, first-/higher-order logic rule mining techniques [19, 23] cannot scale to databases consisting of thousands of records without parallelization [8, 26]. Current techniques usually limit the expressiveness for high performance [9], and this decreases the completeness of induced semantics. Moreover, the selection of best semantics is also challenging, for the number of applicable rules induced from a knowledge base is much larger than required for the summarization.

This paper bridges the gap between RDF KB summarization and first-order logic rule mining. We propose a novel summarization technique based on first-order Horn rules where quantified variables are extensively taken into account. The formal definitions illustrate a new summarization problem: inducing Horn rules from an RDF KB, such that the KB is separated into two parts, where one is inferable (thus removable) by the other with respect to the rules. The top-down rule mining mechanism is also improved to maximize the reusability of cached contents. Contributions of this paper include:

- We are the first to employ first-order Horn rules in RDF KB summarization. Variables are explored to extend the coverage and the completeness of semantic patterns. The new approach is also applicable to relational databases.
- We refine the extension operations in top-down rule mining to a smaller step size, such that the conciseness and performance are both improved.
- We qualitatively analyze the superiority of our approach and demonstrate the reasons with quantitative experimental results. The experiments show that our technique summarizes a database to less than 40% of the size, covering more than 70% contents with induced rules. The performance of our technique is up to two orders faster than the rivals.

The remains of the paper are organized as follows: Sect. 2 reviews major studies in RDF KB summarization. Definitions and details of our approach are proposed in Sect. 3. Section 4 evaluates the performance of our technique and shows evidence of the improvement from a quantitative perspective. Finally, Sect. 5 concludes the entire paper.

2 Related Work

RDF KB summarization aims to extract concise and precise abstraction from facts and ontologies, providing a preview and overall understanding of large-scale knowledge data. Structural, statistical, and pattern/rule-based approaches have been studied for over a decade.

Structural approaches represent the summary as a smaller graph, where vertices and edges are either fragments of the original graph or converted according to some mapping criteria. *Quotient Graphs* [25] are widely applied in many structural approaches. Vertices in a quotient graph represent collections of vertices in the original graph according to an equivalence relation over the vertices. An edge in the quotient graph represents shared edges between the adjacent vertex collections. Forward and backward (bi)simulation [7] properties guarantee that a query on the quotient summary of a knowledge graph returns non-empty results if the results are non-empty from the original database. Indexing [16] is the major benefit of the structural approaches.

Statistical approaches focus on quantitative summaries for visualization [6], query answering [22], selective data access [13], and description generation [10]. The approaches are motivated by the source selection problem, where quantitative statistics reports on how relevant a knowledge base is to a query [13]. Query sensitive information, such as the existence or quantity of relevant entities, triples, or schematic rules, is calculated and stored [22]. To evaluate the relevance to a topic, centrality and frequency analyses within a neighborhood are employed to entities and ontological schema [10]. Summarization techniques for other data types, such as text, are also used to rank objects in different circumstances [18].

Pattern/rule-based approaches employ data mining approaches to extract frequent patterns, in the form of graph structures or rules, from the RDF graph. [28] summarizes with a set of approximate graph patterns in accordance with SPARQL query evaluations. KGIST [2] encodes RDF graph structures into bit strings and takes advantage of information theory to minimize the description length of the entire bit string. The codebook for bit compression represents sub-structures in the original graph. Meier [17] studied an RDF minimization problem under user-defined constraints via Datalog programs. The constrained minimization problem has been proved intractable, and the author identifies a tractable fragment solvable in polynomial time.

LLC [14] summarizes and compresses Linked Open Data (LOD) via association rules, and Fig. 1 shows the overall workflow. LLC converts an RDF knowledge base into a transactional database, and the itemsets consist of objects or relation-object pairs for every subject in the graph. Then FP-growth [12] is used to extract a list of frequent itemsets, and association rules are ranked and selected according to a measure representing the capability of replacement. The original graph is separated into two parts: G_A and G_D , by matching the rules on each itemset. G_D contains triples that cannot be replaced from the knowledge base, and triples in G_A are the replacement of those in $R(G_A)$. Thus, G_A , G_D , and the set of rules R make up the summary. The recovery of the original KG is

accomplished by applying R on G_A .

Rules (1) and (2) are two types of association rules extracted by LLC:

$$\wedge_{i=1}^n \langle X, p, o_i \rangle \leftarrow \langle X, p, o \rangle \quad (1)$$

$$\wedge_{i=1}^n \langle X, p_i, o_i \rangle \leftarrow \langle X, p, o \rangle \quad (2)$$

where $\langle s, p, o \rangle$ refers to a triple, p, p_i are relations, o, o_i are entities, and X is a universally quantified variable. The above rules can be converted to the following Horn rules:

$$p(X, o_i) \leftarrow p(X, o), i = 1, \dots, n \quad (3)$$

$$p_i(X, o_i) \leftarrow p(X, o), i = 1, \dots, n \quad (4)$$

Inductive Logic Programming (ILP) provides top-down [9, 23, 27] and bottom-up [19] solutions to logic rule mining. Probability can also be used for noise tolerance [24]. Parallelization is often employed when inducing from large-scale databases [8, 26].

3 Summarization via First-Order Horn Rules

This section presents the formal definition of the summarization with first-order Horn rules and shows how the rules are used in the solution framework. The logic rule mining process is also improved to extensively explore quantified variables and maximize the reusability of cached contents. The advantages of our technique are demonstrated by comparisons against LLC.

3.1 Preliminaries, Definitions and Notation Conventions

Let Σ be a finite set of constant symbols, e.g. $\{a, b, c, \dots\}$. Let Γ be a finite set of variable symbols, e.g. $\{X, Y, Z, \dots\}$. Let $\mathcal{P}^n (n \geq 0)$ be a finite set of n -ary predicate symbols (i.e. relations), and $\mathcal{P} = \bigcup_{i \geq 0} \mathcal{P}^i$. A **first-order predicate** (or simply, predicate) is composed of a predicate symbol and a list of arguments enclosed in parentheses, written as $p(t_1, \dots, t_k)$, where $p \in \mathcal{P}^k, t_i \in \Sigma \cup \Gamma$. Let P be a predicate, $\phi(P)$ is the **arity** of P . P is a **ground predicate** if all arguments are constants. The above definitions do not break those in First-order Predicate Logic (FOL). In the context of RDF knowledge bases, all predicate symbols are binary, although the formalization and solution to the summarization problem fit in the broader domain of the relational data model.

Formally, an RDF knowledge base is a finite set of binary ground predicates. In FOL, the truth value of a ground predicate is determined by the interpretation and domain. In this paper, the interpretation of non-logic symbols is the

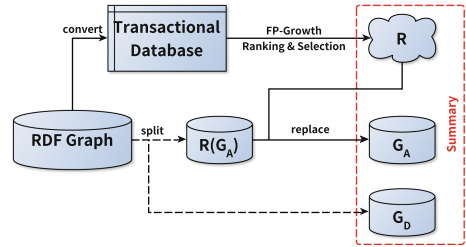


Fig. 1. LLC workflow

definition of relations in databases, and the domain is the set of relation names and constant values. Therefore, a ground predicate P is *True* according to some database \mathcal{D} if and only if $P \in \mathcal{D}$.

A **first-order Horn rule** is of the form: $Q \leftarrow \wedge_i P_i$, where Q, P_i are **atoms** (predicates or the negations). In this paper, only non-negative atoms are considered in the rules. Q is called the **head** of the rule, and predicates P_i make up the **body**. Q is entailed by P_i if P_i are all *True*, that is, $(\wedge_i P_i) \wedge (Q \leftarrow \wedge_i P_i) \models Q$. Thus, by binding the variables in the entailment, the grounded predicate Q' is entailed by grounded predicates P'_i w.r.t. the rule r and a database \mathcal{D} if every $P'_i \in \mathcal{D}$, written as $\{P'_i\} \models_r Q'$. Let \mathcal{S}, \mathcal{T} be sets of ground predicates, \mathcal{H} be a set of first-order Horn rules, $\mathcal{S} \models_{\mathcal{H}} \mathcal{T}$ if $\forall T \in \mathcal{T}, \exists \mathcal{S}' \subseteq \mathcal{S}, r \in \mathcal{H}$, such that $\mathcal{S}' \models_r T$. Suppose T is entailed by a set of predicates w.r.t. a rule r . If $T \in \mathcal{D}$, T is said to be **positively entailed** by \mathcal{S} w.r.t. r ; otherwise, T is **negatively entailed**. If a predicate is positively entailed by some grounding of r , the grounding is called an **evidence** of the predicate. The set of positive and negative entailments w.r.t. rule r is denoted by \mathcal{E}_r^+ and \mathcal{E}_r^- , and $\mathcal{E}_r = \mathcal{E}_r^+ \cup \mathcal{E}_r^-$.

Notation Conventions. Capital letters refer to variables, such as X, Y . **Unlimited Variables (UVs)** are variables assigned to only one argument in some rule; **Limited Variables (LVs)** are those assigned to at least two arguments. A question mark ('?') refers to a unique UV in a rule. Uncapitalized words as arguments refer to constants, e.g., *tom*. Uncapitalized words before the parenthesis or a period are predicate symbols, and the number after the period is the index of the argument in the predicate, starting from 0, such as *father.0*. For example, the following two rules are the same. Variables X and Y are LVs, while Z and W are UVs and can be simplified to two question marks.

$$p(X, Y, Z) \leftarrow q(X, Y), s(Y, W) \quad (5)$$

$$p(X, Y, ?) \leftarrow q(X, Y), s(Y, ?) \quad (6)$$

Definition 1 (Knowledge Graph Summarization). Let \mathcal{D} be an RDF KB. The summarization on \mathcal{D} is a triple $(\mathcal{H}, \mathcal{N}, \mathcal{C})$ with minimal size, where \mathcal{H} (for “Hypothesis”) is a set of inference rules, both \mathcal{N} (for “Necessaries”) and \mathcal{C} (for “Counterexamples”) are sets of predicates. $\mathcal{D}, \mathcal{H}, \mathcal{N}, \mathcal{C}$ satisfies: 1) $\mathcal{N} \subseteq \mathcal{D}$; 2) $\mathcal{N} \models_{\mathcal{H}} (\mathcal{D} \setminus \mathcal{N}) \cup \mathcal{C}$; 3) $\forall e \notin \mathcal{D} \cup \mathcal{C}, \exists r \in \mathcal{H}, \mathcal{N} \models_r e$.

The size of $(\mathcal{H}, \mathcal{N}, \mathcal{C})$ is $\|\mathcal{H}\| + |\mathcal{N}| + |\mathcal{C}|$. $|\mathcal{N}|$ is the number of predicates in \mathcal{N} , and so be $|\mathcal{C}|$. $\|\mathcal{H}\|$ is defined as the sum of lengths of all rules in it.

In LLC, the total size of the patterns is the number of rules—that is—the length of each rule is one, no matter what form and pattern it describes. The coarse definition does not reflect the complexity of identifying semantic patterns. Other rule mining studies [11, 23, 27] intuitively count in the number of terms or different variables, which emphasizes to some extent the complexity of identifying a pattern, but it is still not convincing enough.

In our technique, the length of a Horn rule is measured by the total arity of the rule and the number of different variables at the same time:

$$|r| = \left(\sum_{P \in r} \phi(P) \right) - var(r) \quad (7)$$

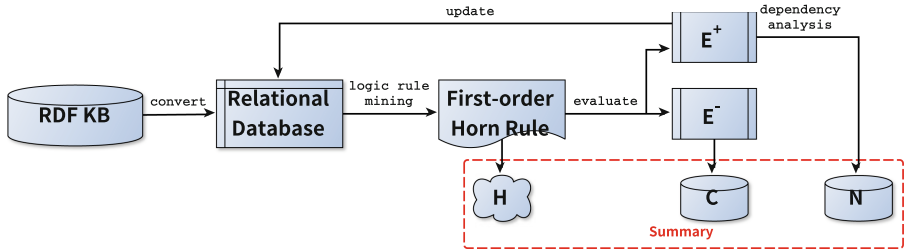


Fig. 2. The workflow of our technique

Table 1. An example knowledge base

<i>s</i>	<i>p</i>	<i>o</i>	<i>s</i>	<i>p</i>	<i>o</i>	<i>s</i>	<i>p</i>	<i>o</i>
tom	father	jerry	tom	gender	male	tom	type	man
bob	father	alice	bob	gender	male	bob	type	man
matt	father	adam	matt	gender	male	matt	type	man
daniel	father	felix	daniel	gender	male	felix	type	man

$var(r)$ is the number of different variables in rule r . Intuitively speaking, the above definition reflects the *minimum* number of equivalence conditions that identify the pattern in r . For example, the length of Rule (8) is 2, because the pattern is characterized by two conditions: $gender.0 = father.0$, $gender.1 = male$. The UV in Rule (8) is existentially quantified.

The size of every triple in Definition 1 is one no matter what relation and entity are represented by the triple because the logic-rule-based summarization cover and remove each triple as a whole. The comparison (Fig. 4a) between summarization ratios and compression ratios has justified that it is proper to define the size of a triple as one.

3.2 Summarization Workflow and the Recovery

Figure 2 shows the overall workflow of our technique. An RDF KB is converted to a relational database, where the subjects and objects are the two arguments in the relations. Each triple in the KB is converted to a single record in the relational database. Labels and types are converted to unary relations where relation names are from the label or type value. Then, logic rules are iteratively induced from each relation until no proper rule is returned. Each Horn rule is evaluated on the database to find the entailments and corresponding evidence. Negatively entailed records are simply collected in the counterexample set \mathcal{C} . Positively entailed records and the corresponding evidence are further analyzed to finally determine the set \mathcal{N} , in case that there are circular entailments in the summarization.

Table 2. Converted Relational Database of Table 1

tom	jerry	tom	male	tom
bob	alice	bob	male	bob
matt	adam	matt	male	matt
daniel	felix	daniel	male	felix

Table 1 shows an example RDF knowledge base, and Table 2 shows the converted version. The original size of the KB is 12, and the following rules are induced for the summarization:

$$gender(X, male) \leftarrow father(X, ?) \quad (8)$$

$$man(X) \leftarrow father(X, ?) \quad (9)$$

The total size of the rules is 3, and only 1 counterexample $man(daniel)$ is generated by the rules. The 5 records in bold font remain in \mathcal{N} , and the others can be entailed from \mathcal{N} w.r.t. the above rules thus are removable. Therefore, the total size of the summarization is 9.

The evidence of positively entailed triples can be represented as a graph, where edges refer to the inference dependency from the body to the head. Therefore, circular dependencies occur as directed cycles in the graph. Figure 3 shows an example where the following rules are included in \mathcal{H} and cause the cycles:

$$man(X) \leftarrow gender(X, male)$$

$$gender(X, male) \leftarrow man(X)$$

Fig. 3. An example of circular entailment

Minimum Feedback Vertex Set (MFVS) [4] algorithms can be used to break the cycles and vertices in the MFVS solution should also be included in \mathcal{N} to make sure every removed record is inferable.

The recovery is simple in our technique: given that the circular dependencies are resolved in the summarization, all removed records can be regenerated by iteratively evaluating Horn rules in \mathcal{H} until no record is added to the database. Counterexamples should be excluded to keep data consistency.

Association rules adopt a limited number (usually one [14] or two [21]) of universally quantified variables, and the patterns are only expressed by the co-occurrences of entities. Thus, general correlations represented by more variables and existential quantifiers are not captured. In the above example, the only inducible association pattern by LLC is the following (or the reverse):

$$type(X, man) \leftarrow gender(X, male) \quad (10)$$

The triples in relation **gender** (or **type**) are not removable. Thus, the conciseness, coverage, and semantic completeness are low in LLC, even though part of the schematic overview has been correctly induced from the data. It is possible to hardcode various semantics into different association structures, such as varying the variables from the subject to the object or even the relation. However, the structures rely on human input and are often tedious to enumerate.

3.3 First-Order Horn Rule Mining

The most critical component in our technique is the induction of first-order Horn rules. Logic rule mining has been extensively studied in the Inductive Logic Programming (ILP) community [5]. Both top-down and bottom-up methodologies have been proposed and optimized for over three decades. The bottom-up strategy regards facts as specific-most rules and merges correlated ones to generalize [19]. The top-down strategy operates in the inverse direction, where rules are constructed from general to specific by imposing new restrictions on candidates [23, 27]. Top-down mining techniques are easier to understand and optimize and are employed in more knowledge-based applications.

Our technique also follows the top-down methodology, and the specialization is refined to improve performance. In previous works, such as FOIL [23] and AMIE [9], candidate rules are specialized by simply appending new atoms to the body of Horn rules. The specialization in the pattern semantics is not well-organized because some newly imposed conditions are repeatedly applied to the candidates, and the number of applicable predicates in each step of specialization is exponential to the maximum arity of the relations if inducing on relational databases. In our approach, a candidate rule is extended in a smaller step size which corresponds to the equivalence between a column and another or a constant value. For example, Rule (8) is constructed in the following order:

$$\begin{aligned} & \textit{gender}(?, ?) \leftarrow \\ & \textit{gender}(X, ?) \leftarrow \textit{father}(X, ?) \\ & \textit{gender}(X, \textit{male}) \leftarrow \textit{father}(X, ?) \end{aligned}$$

The benefit of this modification is three-fold: 1) The extension operations are feasible to relations of arbitrary arities without increasing the difficulty of enumerating applicable predicates to the body. The number of applicable extensions is polynomial to the rule length and the arity of relations. 2) The small-step exploration employs existentially quantified variables with lower cost compared to current logic rule mining techniques, no mention of the association ones. 3) The specialization maximizes the reusability of intermediate results and is better cooperated with caching techniques in relational databases, such as materialization. The reason is that the specialization by each newly imposed condition is updated and stored only once during the induction. Together with pruning [27] and parallelization techniques [8, 26], the performance of logic rule mining will no longer be the stopping reason for RDF KB summarization.

Searching for the best logic rule is accomplished with the beam search, similar to the FOIL system, except that an RDF KB does not provide negative examples. Therefore, the Closed World Assumption (CWA) is adopted in our technique to enumerate the negative examples if necessary. The quality of a Horn rule r is measured by the reduction of overall size:

$$\delta(r) = |\mathcal{E}_r^+| - |\mathcal{E}_r^-| - |r| \quad (11)$$

Table 3. Dataset overview

Datasets	Short	#Rel.	#Entity	#Triple	#Label
Elti	E	10	47	318	–
Family.simple	Fs	4	82	322	1
Dunur	D	17	26	466	–
DBpedia.factbook	DBf	2	335	880	Default
Family.medium	Fm	9	142	1242	1
Student Loan	S	9	1031	6317	–
UMLS	U	46	135	6664	Default
WN18	WN	18	41K	193K	Default
NELL	N	1083	44K	278K	821
FB15K	FB	1345	15K	607K	Default

4 Evaluation

This section evaluates our technique and answers the following research questions:

- Q1 To what extent are RDF KBs summarized by first-order Horn rules?
 Q2 How and why does our technique outperform state-of-the-art methods?
 Q3 How fast does our technique induce logic rules?

Datasets. We use ten open-access datasets, without deliberate selection, from various domains, including relational databases, fragments of popular knowledge graphs that are widely used as benchmarks, and two synthetic datasets. Table 3 shows statistics of these datasets.¹ “E”, “D”, and “S” are relational databases themselves, and the others are converted to the corresponding relational form. Given that KGIST requires entity labels in databases, we assign a default label to datasets where the label information is unavailable. Datasets tested in LLC are outdated and no longer accessible thus are not used in our tests. The datasets are not extremely large because FOIL and KGIST are not implemented in a parallel manner, and we compare the speed in a single thread mode to demonstrate the impact of the small-step specialization operations. More importantly, the datasets are sufficient to emphasize the superiority of our technique.

Rivals and Settings. We compare our approach against four state-of-the-art techniques: FOIL, LLC, AMIE, and KGIST. The summarization quality is compared mainly against LLC. KGIST and AMIE are also compared for summarization, as KGIST is devoted to the same purpose via a graph-based approach, and AMIE can be slightly modified, for a fair comparison, to summarize KGs by selecting the rules useful for reducing the overall size. FOIL and AMIE are chosen as the competitors for speed comparison, as both of them induce first-order

¹ E, D, S are available at: <https://relational.fit.cvut.cz/>; Fm, Fs are synthetic, and the generators are available with the project source code.

Horn rules and are the most similar to ours. However, neither the source code nor compiled tool is available for LLC. Therefore, we reimplemented the algorithm according to the instructions in [14]. The latest version of AMIE, AMIE3, is used in the experiment, and Partial Completeness Assumption (PCA) is employed in AMIE. Our technique is implemented in Java 11 and is open-source on GitHub². All tests were carried out in a single thread on Deepin Linux (kernel: 5.10.36-amd64-desktop) with Ryzen 3600 and 128 GB RAM. The beamwidth for our technique is 5.

Metrics. The quality of summarization is quantitatively reported by the summarization ratio (θ), pattern/rule complexity ($|r|$) and connectivity (ρ), and knowledge coverage (τ). The summarization ratio is defined as:

$$\theta(\mathcal{D}) = (|\mathcal{H}| + |\mathcal{N}| + |\mathcal{C}|)/|\mathcal{D}| \quad (12)$$

where $|\mathcal{H}|$, $|\mathcal{N}|$, $|\mathcal{C}|$, and $|\mathcal{D}|$ in LLC, AMIE, and our technique follow Definition 1. The components in KGIST are measured by the length of bit strings. The connectivity is the connection density in relations and reflects the completeness of exhausting hidden semantics in a knowledge base:

$$\rho(\mathcal{H}) = |\{(p, q) | p, q \in \mathcal{P}, p, q \text{ appear in the same rule } r \in \mathcal{H}\}|/|\mathcal{P}|^2 \quad (13)$$

In our technique, the converted “type” or “label” relations are counted as one single relation, as is calculated in other techniques. The knowledge coverage is the ratio of all inferable (not necessarily removable) triples over the entire set:

$$\tau(\mathcal{D}) = |\{e \in \mathcal{D} | \mathcal{D} \models_{\mathcal{H}} e\}|/|\mathcal{D}| \quad (14)$$

4.1 Summarization with Horn Rules

The results in this section answer Q1: The summarization and compression ratios of our technique are up to 40%; Circular entailments frequently appear in the summarization but are easy to resolve.

Figure 4 shows summarization statistics of our technique on the datasets. Θ refers to the compression ratio measured by input/output files in Bytes. The bars in three different colors in Fig. 4a add up to the total summarization ratio, and it is shown that more than 60% contents are replaced by logic rules in the datasets. Compared to the number of remaining triples, the sizes of rules and counterexamples are negligible. The reason is that there are usually clear topics and themes in modern knowledge bases, and within the topics, some relations extend details of complex concepts. Moreover, necessary redundancies are included for high completeness of domain knowledge, as most facts are automatically extracted from the open-source text and checked by human. For example, the followings are some rules induced from the datasets:

$$part_of(X, Y) \leftarrow has_part(Y, X) \quad (15)$$

$$uncle(X, Y) \leftarrow brother(X, Z), aunt(Z, Y) \quad (16)$$

$$aunt(X, Y) \leftarrow sister(X, Z), uncle(Z, Y) \quad (17)$$

² <https://github.com/TramsWang/SInC>.

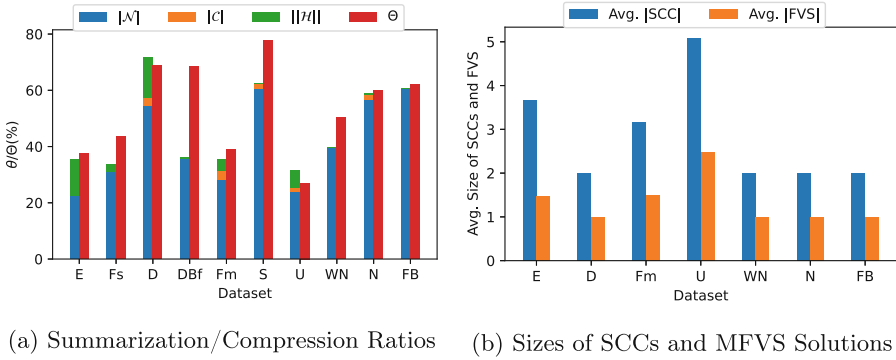


Fig. 4. Summarization detail

Relation `aunt` and `uncle` in dataset “Fm” can be mutually defined by each other with some auxiliary relations. Many relations, such as `part_of` and `has_part` in “WN”, are symmetric, and this is a common circumstance in modern KGs. Figure 4b shows the evidence by counting the sizes of Strongly Connected Components (SCCs) in the graph that represents inference dependencies of triples. The average sizes of SCCs in large-scale KGs, such as “FB”, “WN”, and “N”, are approximately 2, which testifies the above analysis. Moreover, from the figure, we can conclude that the cycles are not large in the datasets and can be efficiently solved by MFVS algorithms, even in a greedy manner.

Figure 4a also shows that our technique successfully applies to relational data-bases. Moreover, the summarization ratio is close to the file compression ratio. It is proper to define the size of a triple as one. θ and $\hat{\theta}$ have an apparent difference in “DBf” because the following induced rule eliminates entities after triples are removed: $sameAs(X, X) \leftarrow$, and extra information for the entities should be recorded for a complete recovery. However, the information is not included in Definition 1, as the above case is rare in practice.

4.2 Quality of Summarizations

In this section, we compare our technique against the state-of-the-art tools: LLC, AMIE, and KGIST, and answer Q2: Our technique induces more expressive logic rules than the state-of-the-art; Rules in our technique cover more triples, reflect more comprehensive semantics, and are more representative.

Figure 5 shows the overall summarization ratios of the techniques. “E”, “D”, and “S” are not compared as the competitors cannot handle relational databases. Our technique outperforms the others in almost all datasets. Some of the ratios by LLC are larger than 100% because many rules are induced but not used to replace triples. For example, the following two rules are induced from “DBf” by

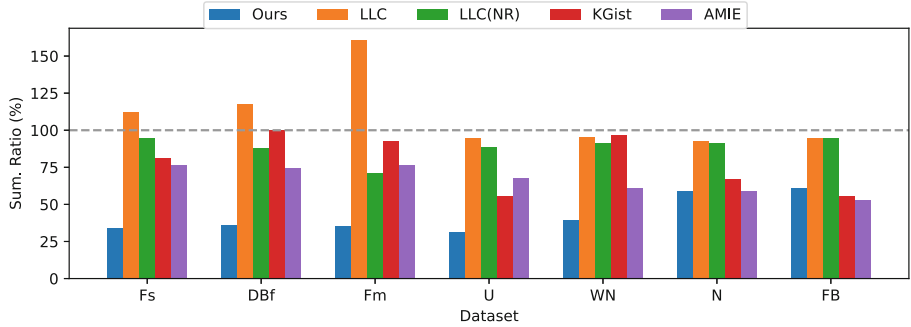


Fig. 5. Summarization comparison

Table 4. Blocked rules (%) in LLC

Dataset	Fs	DBf	Fm	U	WN	N	FB
Blocked rules (%)	52.63	48.39	68.46	60.00	44.23	42.90	45.00

LLC:

$$spokenIn(X, Russia) \wedge type(X, default) \leftarrow spokenIn(X, Kazakhstan) \quad (18)$$

$$spokenIn(X, Russia) \wedge type(X, default) \leftarrow spokenIn(X, Uzbekistan) \quad (19)$$

But Rule (19) is blocked from replacing the head triples if Rule (18) is applied. According to Table 4, about half of the rules in LLC are blocked due to the above reason.

However, excluding the size of rules (shown as “LLC (NR)” in Fig. 5) does not change the fact that LLC is not competitive to logic-rule-based techniques. The main reason is that association patterns are applicable to only a small part of triples in the datasets. For example, Rule (18) is the most frequently used in “DBf”, and it replaces only 18 triples, the proportion of which is only 2.05%, in the dataset. Figure 6a compares the overall coverage of all techniques. The association patterns induced by LLC cover only about 20% triples in a KG. The low coverage is further explained by Fig. 6b. The figure shows that the number of itemsets, i.e., potential association patterns, exponentially decreases with increasing size of the itemset. More importantly, the number is much smaller than the matching arguments, represented by variables. Therefore, the association patterns are not representative as first-order logic rules are.

Figure 6c compares the connectivity (see Eq. (13)) of induced patterns. Given that the connectivity varies a lot in datasets, for a clear illustration, we compare the connectivity of other techniques to ours. Therefore, the red line at value 1.0 denotes the connectivity of our technique, and the others are the relative values. In most cases, our technique induces patterns that correlate the most relations, thus reflecting the semantics more comprehensively in the data. Although LLC combines more relations in “DBf” and “WN”, the average numbers of triples

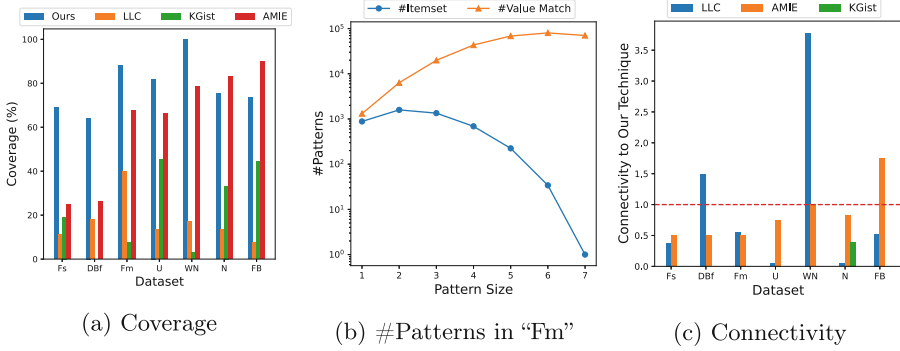


Fig. 6. Pattern comparison

inferable by the rules induced from the two datasets are 2.07 and 11.17, while the numbers for our technique are 124 and 8358.22. Hence, the coverage of our technique remains extensive even though the connectivity is occasionally low.

Figure 7 compares the length of patterns in the dataset “N” according to the measure proposed in Sect. 3.1. The reason why LLC induces longer patterns is that the association patterns consist mainly of entities, each of which is size 1 in the new length measure, while variables represent the matching between arguments with much less cost.

The comparisons with other state-of-the-art techniques also approve that logic-rule-based approaches generalize better than graph-pattern-based ones, thus producing more concise summaries. The summarization ratios for our technique and AMIE are smaller than LLC and KGIST. The knowledge coverage is also significantly more extensive than the graph-pattern-based approaches. Most of the rules in KGIST are at length one because the patterns it describes usually involve a single relation and the direction, and this is also the reason for almost-zero connectivity in KGIST.

Our technique summarizes better than AMIE because rules induced in our technique are longer and contain existentially quantified UVs. For example, the following rule is simple but out of reach of AMIE, because it contains a UV:

$$gender(X, female) \leftarrow mother(X, ?)$$

Moreover, the rule evaluation metric adopted in AMIE is based on PCA, which assumes the functionality of relations in knowledge bases. However, the PCA in

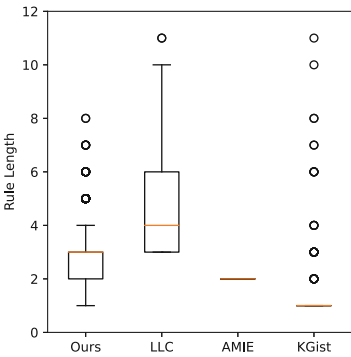


Fig. 7. Rule Lengths on NELL

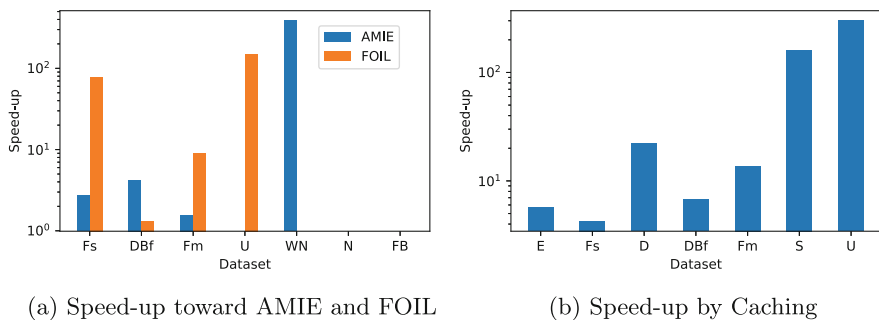


Fig. 8. Rule mining speed comparison

AMIE is not suitable for the summarization purpose. For example, our technique covers triples in relation `produces` with only 10 counterexamples, while AMIE does with 97.

4.3 Rule Mining Speed

The results in this section answer Q3: The speed of our technique is up to two orders faster than FOIL and AMIE, and the speed-up is mainly due to the small-step specialization operations together with caching.

Both AMIE and FOIL induce first-order Horn rules and are the most similar techniques to ours. AMIE restricts the length and applicable variables in the rules, and it runs in multi-threads. The maximum length and the number of threads in AMIE should be set to 5 and 1 to compare the performance under approximately equal expressiveness. However, AMIE frequently ends up with errors under the above setting. The adopted parameters for maximum length and threads are 4 and 3, respectively. Therefore, the actual speed-up is larger than the recorded numbers in Fig. 8a. In the figure, the missing numbers are because of program failures due to program errors or memory issues in FOIL and AMIE.

The results show that our technique performs one to two orders faster than AMIE and FOIL. Although AMIE adopts an estimation metric for heuristically selecting promising specializations of rules, it tends to repeatedly cover triples by different rules. No more than 10% rules produced by AMIE are used in the summarization. Although AMIE employs an in-memory database with combinatorial indices, the caching is not fully explored due to the types of terms it appends to the rules. For example, Rules (21) and (22) are two extensions of Rule (20) in AMIE. The condition $grandfather.0 = father.0$ has been repeatedly imposed on the base rule during the extension.

$$grandfather(X, Y) \leftarrow \tag{20}$$

$$grandfather(X, Y) \leftarrow father(X, Y) \tag{21}$$

$$grandfather(X, Y) \leftarrow father(X, Z) \tag{22}$$

FOIL finds the best description for relations under the metric “Information Gain”. FOIL does not over-explore the search space of Horn rules as AMIE does, but the tables are repeatedly joined, as FOIL does not cache the intermediate result of candidate rules during the construction. Figure 8b shows the speed-up by caching intermediate results, and this explains most of the difference between FOIL and our technique. Moreover, Fig. 8b also shows that the speed-up by caching is more significant in larger datasets.

5 Conclusion

This paper proposes a novel summarization technique on RDF KBs by inducing first-order Horn rules. Horn rules significantly extend the coverage, completeness, and conciseness due to extensive exploration of variables compared to the association and graph-structure patterns. The small-step specialization operations also improve the performance of rule induction by maximizing the reusability of cached contents. As shown in the experiments, our technique summarizes KBs to less than 40% of the original size, covers more than 70% triples, and is up to two orders faster than the rivals. Our technique not only produces a concise and faithful summary of RDF KBs but is also applicable to relational databases. Therefore, the new technique is practical for a broader range of knowledge-based applications.

References

1. Ahmed, M.: Data summarization: a survey. *Knowl. Inf. Syst.* **58**(2), 249–273 (2019)
2. Belth, C., Zheng, X., Vreeken, J., Koutra, D.: What is normal, what is strange, and what is missing in a knowledge graph: Unified characterization via inductive summarization. In: *The Web Conference (WWW)* (2020)
3. Čebirić, Š, et al.: Summarizing semantic graphs: a survey. *VLDB J.* **28**(3), 295–327 (2019)
4. Chen, J., Liu, Y., Lu, S., O’sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pp. 177–186 (2008)
5. Cropper, A., Dumancic, S., Muggleton, S.H.: Turning 30: New ideas in inductive logic programming. In: *IJCAI* (2020)
6. Dudáš, M., Svátek, V., Mynarz, J.: Dataset summary visualization with LODSight. In: Gandon, F., Guéret, C., Villata, S., Breslin, J., Faron-Zucker, C., Zimmermann, A. (eds.) *ESWC 2015. LNCS*, vol. 9341, pp. 36–40. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25639-9_7
7. Fan, W., Li, J., Wang, X., Wu, Y.: Query preserving graph compression. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 157–168 (2012)
8. Fonseca, N.A., Srinivasan, A., Silva, F., Camacho, R.: Parallel ILP for distributed-memory architectures. *Mach. Learn.* **74**(3), 257–279 (2009)

9. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with AMIE+. *VLDB J.* **24**(6), 707–730 (2015)
10. Gunaratna, K., Thirunarayan, K., Sheth, A.: Faces: diversity-aware entity summarization using incremental hierarchical conceptual clustering. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
11. Hammer, P.L., Kogan, A.: Quasi-acyclic propositional horn knowledge bases: optimal compression. *IEEE Trans. Knowl. Data Eng.* **7**(5), 751–762 (1995)
12. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. *ACM SIGMOD Rec.* **29**(2), 1–12 (2000)
13. Hose, K., Schenkel, R.: Towards benefit-based RDF source selection for SPARQL queries. In: Proceedings of the 4th International Workshop on Semantic Web Information Management, pp. 1–8 (2012)
14. Joshi, A.K., Hitzler, P., Dong, G.: Logical Linked data compression. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) *ESWC 2013*. LNCS, vol. 7882, pp. 170–184. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38288-8_12
15. Kushk, A., Kochut, K.: Esdl: Entity summarization with deep learning. In: The 10th International Joint Conference on Knowledge Graphs, pp. 186–190 (2021)
16. Luo, Y., Fletcher, G.H., Hidders, J., Wu, Y., De Bra, P.: External memory k-bisimulation reduction of big graphs. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 919–928 (2013)
17. Meier, M.: Towards rule-based minimization of RDF graphs under constraints. In: Calvanese, D., Lausen, G. (eds.) *RR 2008*. LNCS, vol. 5341, pp. 89–103. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88737-9_8
18. Motta, E., et al.: A novel approach to visualizing and navigating ontologies. In: Aroyo, L., et al. (eds.) *ISWC 2011*. LNCS, vol. 7031, pp. 470–486. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25073-6_30
19. Muggleton, S.H., Lin, D., Pahlavi, N., Tamaddoni-Nezhad, A.: Meta-interpretive learning: application to grammatical inference. *Mach. Learn.* **94**(1), 25–49 (2014)
20. Palmonari, M., Rula, A., Porrini, R., Maurino, A., Spahiu, B., Ferme, V.: ABSTAT: linked data summaries with ABstraction and STATistics. In: Gandon, F., Guéret, C., Villata, S., Breslin, J., Faron-Zucker, C., Zimmermann, A. (eds.) *ESWC 2015*. LNCS, vol. 9341, pp. 128–132. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25639-9_25
21. Pan, J.Z., Pérez, J.M.G., Ren, Y., Wu, H., Wang, H., Zhu, M.: Graph pattern based rdf data compression. In: Supnithi, T., Yamaguchi, T., Pan, J.Z., Wuwongse, V., Buranarach, M. (eds.) *JIST 2014*. LNCS, vol. 8943, pp. 239–256. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15615-6_18
22. Pires, C.E., Sousa, P., Kedad, Z., Salgado, A.C.: Summarizing ontology-based schemas in pdms. In: 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010), pp. 239–244. IEEE (2010)
23. Quinlan, J.R.: Learning logical definitions from relations. *Mach. Learn.* **5**(3), 239–266 (1990)
24. Raedt, L.D., Kersting, K.: Statistical relational learning. In: Sammut, C., Webb, G.I. (eds.) *Encyclopedia of Machine Learning*, pp. 916–924. Springer (2010). https://doi.org/10.1007/978-0-387-30164-8_786
25. Sanders, P., Schulz, C.: High quality graph partitioning. *Graph Partition. Graph Cluster.* **588**(1), 1–17 (2012)
26. Srinivasan, A., Faruque, T.A., Joshi, S.: Data and task parallelism in ILP using mapreduce. *Mach. Learn.* **86**(1), 141–168 (2012)

27. Zeng, Q., Patel, J.M., Page, D.: Quickfoil: scalable inductive logic programming. *Proc. VLDB Endow.* **8**(3), 197–208 (2014)
28. Zneika, M., Lucchese, C., Vodislav, D., Kotzinos, D.: Summarizing linked data RDF graphs using approximate graph pattern mining. In: *EDBT 2016.*, pp. 684–685 (2016)