# Exception-enriched Rule Learning from Knowledge Graphs (Extended Abstract)

Mohamed Gad-Elrab[a], Daria Stepanova[a], Jacopo Urbani[b], Gerhard Weikum[a]

[a]Max Planck Institute of Informatics, Germany
[b]Vrije Universiteit Amsterdam, The Netherlands
{gadelrab,dstepano,weikum}@mpi-inf.mpg.de, jacopo@cs.vu.nl

## 1 Introduction

Recent developments in information extraction have enabled the construction of huge Knowledge Graphs (KGs), e.g., DBpedia [1] or YAGO [8]. To complete and curate modern KGs, inductive logic programming and data mining methods have been introduced to identify frequent data patterns, e.g., *"Married people live in the same place"*, and cast them as rules like $r1 : livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Z)$. These rules can be used for various purposes: First, since KGs operate under Open World Assumption (OWA – i.e. absent facts are treated as unknown), they can be applied to derive new potentially true facts. Second, rules can be used to eliminate erroneous information from the KG.

Existing learning methods restrict to Horn rules [4] (i.e. rules with only positive body atoms), which are insufficient to capture more complex patterns, for instance like $r2: livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Z), not\ researcher(Y)$, i.e., non-monotonic rules. While $r1$ generally holds, the additional knowledge that $Y$ is a researcher could explain why few instances of $isMarriedTo$ do not live together; this can prevent inferring the missing living place by only relying on the $isMarriedTo$ relations.

Thus, for KG completion and curation, understanding exceptions is crucial. While learning non-monotonic rules under Closed World Assumption (CWA – i.e. absent facts are treated as false) is a well-studied problem that lies at the intersection of inductive and abductive logic programming (e.g., [11]), it has not been yet investigated in the context of KGs treated under OWA, despite evident importance of this research direction. To overcome the limitations of prior work on KG rule mining, our goal is to develop methods for learning *non-monotonic rules* from KGs.

We formulate this ambitious task as a version of a theory revision problem [10], where, given a KG and a set of (previously learned) Horn rules, the aim is to update them to nonmonotonic rules, so that their quality is better than the Horn rules'. In [9], we made a first step towards tackling this problem by providing an approach of stepwise rule revision, where novel ranking functions are used to quantify the strength of nonmonotonic rules w.r.t the KG. We did not merely estimate the quality of individual rules in isolation, but considered their cross-talk through a new technique that we call *partial materialization*. We implemented a prototype of our approach and reported on the improvements we obtained both in terms of rules' quality as well as predicted fact quality when performing KG completion. In the remaining of this paper, we summarize the main results from [9] and discuss possible extensions to more general settings.

## 2 Nonmonotonic Rule Learning from Knowledge Graphs

**Problem Statement**. On the Web, knowledge graphs (KG) $\mathcal{G}$ are often encoded using the RDF data model, which represents the content of the graph with a set of triples of the form $\langle subject\ predicate\ object \rangle$. These triples can be seen as positive unary and binary facts, i.e., the above triple corresponds to $object(subject)$ if $predicate = isA$ and to $predicate(object, subject)$ otherwise[1]. KGs are naturally treated under the OWA.

In this work, we focus on non-monotonic rules. A *nonmonotonic logic program* is a set of rules of the form $a_1 \leftarrow b_1, \ldots, b_k\ not\ \ b_{k+1}, \ldots, b_n$ where each $a_i$ and $b_j$ is a first-order atom and $not$ is called negation as failure (NAF) or default negation. The answer set semantics [5] for nonmonotonic logic programs is based on the CWA. Given a ruleset $\mathcal{R}$ and a set of facts $\mathcal{G}$, the models (aka. answers sets) of the program $\mathcal{R} \cup \mathcal{G}$ can be determined following [5]. They reflect the information that can be deduced from $\mathcal{R} \cup \mathcal{G}$ under the answer set semantics.

Let $\mathcal{G}^a$ be a given (possibly incomplete) KG, and let $\mathcal{G}^i$ be the ideal KG that contains nodes from $\mathcal{G}^a$ and all relations between these nodes that hold in the current state of the world. Our ultimate goal is to automatically extract a set of rules $\mathcal{R}$ from $\mathcal{G}^a$, applying which (i.e. computing some answer set of $\mathcal{R} \cup \mathcal{G}^a$) we can obtain the graph $\mathcal{G}^a_{\mathcal{R}}$, which minimally differs from $\mathcal{G}^i$. Our approach is to first learn a set of Horn rules, and then aim at simultaneously revising them by adding negated atoms to the rule bodies. Since normally, the ideal graph $\mathcal{G}^i$ is not available, in order to estimate the quality of a revised ruleset, we devise two generic quality functions $q_{rm}$ and $q_{conflict}$, that take as input a ruleset $\mathcal{R}$ and a KG and output a real value, reflecting the suitability of $\mathcal{R}$ for data prediction. More specifically,

$$q_{rm}(R, \mathcal{G}) = \frac{\sum_{r \in \mathcal{R}} rm(r, \mathcal{G})}{|\mathcal{R}|}, \tag{1}$$

where $rm$ is some standard association rule measure [2]. To measure $q_{conflict}$ for $\mathcal{R}$, we create an extended set of rules $\mathcal{R}^{aux}$, which contains each revised rule in $\mathcal{R}$ together with its auxiliary version. For each rule $r$ in $\mathcal{R}$, its auxiliary version $r^{aux}$ is constructed by: i) transforming $r$ into a Horn rule by removing $not$ from negated body atoms, and ii) replacing the head predicate $a$ of $r$ with a newly introduced predicate $not\_a$ which intuitively contains instances which are *not* in $a$. Formally, we define $q_{conflict}$ as follows

$$q_{conflict}(\mathcal{R}_{NM}, \mathcal{G}) = \sum_{p \in pred(\mathcal{R}^{aux})} \frac{|\boldsymbol{c} \mid p(\boldsymbol{c}), not\_p(\boldsymbol{c}) \in \mathcal{G}_{\mathcal{R}^{aux}}|}{|\boldsymbol{c} \mid not\_p(\boldsymbol{c}) \in \mathcal{G}_{\mathcal{R}^{aux}}|} \tag{2}$$

We are now ready to state our problem: Given a KG $\mathcal{G}$, a set of nonground Horn rules $\mathcal{R}_H$ mined from $\mathcal{G}$, and a quality function $rm$, our goal is to find a set of rules $\mathcal{R}_{NM}$ obtained by adding negated atoms to $Body(r)$ for some $r \in \mathcal{R}_H$ s.t. (i) $q_{rm}(\mathcal{R}_{NM}, \mathcal{G})$ is maximal, and (ii) $q_{conflict}(\mathcal{R}_{NM}, \mathcal{G})$ is minimal.

**Unary rules**. In [9], we focused on rules with unary atoms. To this end, we transformed binary facts in our initial KG to unary ones via propositionalization. Our approach proceeds in four steps.

[1] For simplicity in this work we identify a given graph with its factual representation.

**Step 1.** After mining Horn rules using an off-the-shelf algorithm (e.g., FPGrowth [6], [3] or [4]), we compute for each rule the *normal* and *abnormal* instance sets, defined as

**Definition 1 ($r$-(ab)normal instance set).** *Let $\mathcal{G}$ be a KG and, moreover, let $r: a(X) \leftarrow b_1(X), \ldots, b_k(X)$ be a Horn rule mined from it. Then*

- $NS(r, \mathcal{G}) = \{c \mid b_1(c), \ldots, b_k(c), a(c) \in \mathcal{G}\}$ *is an* $r$-normal instance set*;*
- $ABS(r, \mathcal{G}) = \{c \mid b_1(c), \ldots, b_k(c) \in \mathcal{A}, a(c) \notin \mathcal{G}\}$ *is an* $r$-abnormal instance set.

**Step 2.** Intuitively, if the given data was complete, then the $r$-normal and $r$-abnormal instance sets would exactly correspond to instances for which the rule $r$ holds (resp. does not hold) in the real world. Since the KG is potentially incomplete, this is no longer the case and some r-abnormal instances might in fact be classified as such due to data incompleteness. In order to distinguish the "wrongly" and "correctly" classified instances in the $r$-abnormal set, we construct *exception witness sets* ($EWS$), which are defined as follows:

**Definition 2.** *Let $\mathcal{G}$ be a KG and let $r$ be a Horn rule mined from $\mathcal{G}$. An* $r$-exception witness set $EWS(r, \mathcal{G}) = \{e_1, \ldots, e_l\}$ *is a maximal set of predicates, such that*

*(i)* $e_i(c') \in \mathcal{G}$ *for some* $c' \in ABS(r, \mathcal{G})$, $1 \leq i \leq l$ *and*
*(ii)* $e_1(c), \ldots, e_l(c) \notin \mathcal{A}$ *for all* $c \in NS(r, \mathcal{G})$.

**Steps 3 and 4.** After $EWS$s are computed for all rules in $\mathcal{R}_H$, we use them to create potential revisions (Step 3), i.e., from every $e_j \in EWS(r_i, \mathcal{G})$ a revision $r_i^j$ of $r_i$ is constructed by adding a negated atom over $e_j$ to the body of $r_i$. Finally, we determine a concrete revision for every rule, that will constitute a solution to our problem (Step 4). To find such globally best ruleset revision $\mathcal{R}_{NM}$ many candidate combinations have to be checked, which due to the large size of our $\mathcal{G}$ and $EWS$s might be too expensive. Therefore, instead we incrementally build $\mathcal{R}_{NM}$ by considering every $r_i \in \mathcal{R}_H$ and choosing the locally best revision $r_i^j$ for it.

In order to select $r_i^j$, we introduce four special ranking functions: a naive one and three more advanced functions, which exploit the novel concept of *partial materialization* (***PM***). Intuitively, the idea behind it is to rank candidate revisions not based on $\mathcal{G}$, but rather on its extension with predictions produced by other (selectively chosen) rules (grouped into a set $\mathcal{R}'$), thus ensuring a cross-talk between the rules. We now describe the ranking functions in more details.

- ***Naive*** ranker is the most straightforward ranking function. It prefers the revision $r_i^j$ with the highest value of $rm(r_i^j, \mathcal{G})$ among all revisions of $r_i$.
- ***PM*** ranking function prefers $r_i^j$ with the highest value of

$$\frac{rm(r_i^j, \mathcal{G}_{\mathcal{R}'}) + rm(r_i^{j\,aux}, \mathcal{G}_{\mathcal{R}'})}{2} \tag{3}$$

  where $\mathcal{R}'$ is the set of rules $r_k'$, which are rules from $\mathcal{R}_H \backslash r_i$ with all exceptions from $EWS(r_k, \mathcal{G})$ incorporated at once. Informally, $\mathcal{G}_{\mathcal{R}'}$ contains only facts that can be safely predicted by the rules from $\mathcal{R}_H \backslash r_i$, i.e., there is no evident reason (candidate exceptions) to neglect their prediction.

– ***OPM*** is similar to ***PM***, but the selected ruleset $\mathcal{R}'$ contains only those rules whose Horn version appears above the considered rule $r_i$ in the ruleset $\mathcal{R}_H$, ordered (***O***) based on some chosen measure (e.g., the same as $rm$).
– ***OWPM*** is the most advanced ranking function. It differs from ***OPM*** in that the predicted facts in $\mathcal{G}_{\mathcal{R}'} \setminus \mathcal{G}$ inherit weights (***W***) from the rules that produced them, and facts in $\mathcal{G}$ get the highest weight. These weights are taken into account when computing the value of 3. If the same fact is derived by multiple rules, we store the highest weight. To avoid propagating uncertainty through rule chaining when computing weighted partial materialization of $\mathcal{G}$ we keep predicted facts (i.e., derived by applying rules from $\mathcal{R}'$) separately from the explicit facts (i.e., those in $\mathcal{G}$), and infer new facts using only $\mathcal{G}$.

**Extension to binary rules**. A natural direction for extending the work from [9] is to consider rules involving binary atoms. In this case, there can be a potentially larger number of possible $EWS$ sets to construct and consider. More specifically, if a rule has $n$ distinct variables, then there could be $n+\binom{n}{2}$ candidate $EWS$ sets. Given the large size of KGs, computing all exceptions in every $EWS$ set might be inpractical for scalability reasons. To overcome this issue, the language bias of possible exception candidates should be carefully fixed. Practically, several possibilities for such restriction exist. For instance, one could search only for binary (resp. unary) exceptions, or only consider $EWS$s w.r.t. to the variables in (a certain position of) the head atom. An in-depth analysis of these possibilities is planned for our future work.

## 3 Evaluation

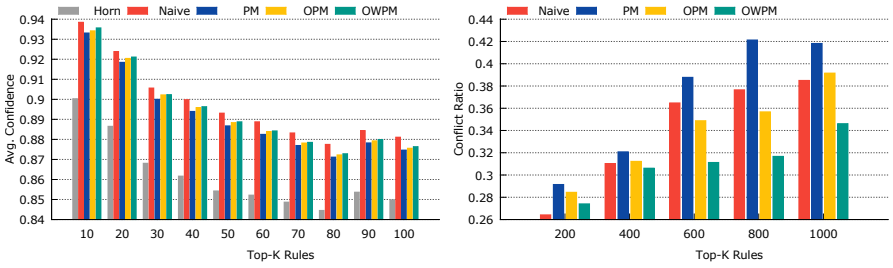
We briefly discuss some of experimental results that are reported in more detail in [9].

**Step 1**. Initially, we considered the Horn rules produced by AMIE [4]. However, they involve unsupported binary predicates and the only unary rules regard the *isA* predicate, which was too limiting for us. Therefore, we used the standard mining algorithm FP-Growth [6] offered by SPMF Library[2] and extracted Horn rules from two well-known KGs: YAGO (general purpose) and IMDB (domain-specific). Before learning Horn rules, we preprocessed a given KG by converting binary facts $predict(subject, object)$ into unary ones $predict\_object(subject)$, and automatically abstracting the new unary predicates using the type hierarchy of the KG to make them more dense and allow mining expressive data patterns. In order to avoid over-fitting, we applied some restrictions to the rules (e.g. we limited to rules with at most four body atoms, a single head atom, a minimum support of $0.0001 \times$ *# entities*, etc.).

**Steps 2 and 3**. We implemented a simple inductive learning procedure to calculate the $EWS$s. We could find $EWS$s for about 6K rules mined from YAGO, and 22K rules mined from IMDB. On average, the $EWS$s for the YAGO's rules contained 3 exceptions, and 28 exceptions for IMDB.

**Step 4**. We evaluated the quality of our rule selection procedure in terms of the increase of rules' confidence and the decrease of the number of conflicts introduced by

[2] `http://www.philippe-fournier-viger.com/spmf/`




(a) Confidence for top-k IMDB revised rules

(b) Ratio of conflicts for IMDB rules

Fig. 1: Average rules' confidence and number of conflicts on IMDB KG.

negated atoms. The confidence shows how well the revised rules adhere to the input. The number of conflicts indicates how consistently the revised rules predict the unseen data. Fig. 1 reports the obtained average rules' confidence of original Horn rules and rules revised with our ranking functions on the IMDB dataset (YAGO's follows a similar behaviour [9]). Fig. 1a reports the average confidence of the original Horn rules. For each ranking method, we show the results for the top $10,\ldots,100\%$ rules ranked by lift.

From Fig. 1a, we make two observations. First, we notice that enriching Horn rules with exceptions increases the average confidence (appr. $3.5\%$). Second, as expected, the highest confidence is achieved by the (*Naive*) procedure, as the latter blindly chooses exceptions that maximize confidence, while ignoring the conflict ratio. However, confidence alone is not sufficient to determine the overall rule's quality, and also consistency of the predictions (i.e., $q_{conflict}$ function) should be taken into account.

In order to evaluate $q_{conflict}$, we computed the number of conflicts by executing the revised rules and their corresponding auxiliary versions ($r^{aux}$) on IMDB KG using the DLV tool [7]. The conflict appears whenever both $p(c)$ and $not\_p(c)$ are derived. Fig. 1b reports the ratio between the number of conflicts and negated derived facts. One can observe that *OWPM* and *OPM* produce less conflicts than the *Naive* function in most of the cases. Moreover, the *OWPM* ranking function works generally better than *OPM* and *PM* functions, i.e., taking into account weights of the predicted facts leads to improved revisions. For instance, for IMDB, moving from *OPM* to *OWPM* reduced the number of conflicts from 775 to 685 on a base of about 2000 negated facts.

In another experiment, we counted the number of derivations that our exceptions prevented using the top-1000 YAGO rules. With the original Horn rules, the reasoner inferred about $924$K new facts, while the revised rules deduced around $890$K facts. In order to assess whether the $34$K predictions neglected due to our revision method are actually erroneous, we sampled 259 random facts from the removed set (we selected three facts for each binary predicate to avoid skewness), and manually consulted online resources (mainly Wikipedia) to determine whether they were indeed incorrect. We found that $74.3\%$ of them were actually faulty predictions. This number provides a first empirical evidence that our method is capable of detecting good exceptions, and hence can improve the general predictive quality of the Horn rules. Unfortunately, since KGs follow OWA, automatic evaluation of predictions is problematic, and human judgment

is often required to estimate the validity of exceptions. Cross validation methods could be adapted for our needs and exploited for evaluation purposes to some extent. This is planned for future work. However, since fully complete versions of the real-world KGs (i.e., $\mathcal{G}^i$) are not available, to measure how correct and probable our exceptions actually are, manual assessment might be still required.

## 4 Discussion and Outlook

We have presented a method for mining nonmonotonic rules from KGs: First learning a set of Horn rules and then revising them by adding negated atoms into their bodies. We evaluated it with various configurations on both general-purpose and domain-specific KGs and observed significant improvements over a baseline Horn rule mining.

Apart from extensions to rules with predicates of higher arity, there are other future directions to explore. First, one can look into extracting evidence for or against exceptions from text and web corpora. Second, our framework can be enhanced by partial completeness assumptions for certain predicates (e.g., all countries are available in the KG) or constants (e.g., knowledge about Barack Obama is complete). We believe these are important research topics that should be studied in the future.

## 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: Proc. of ISWC. pp. 722–735 (2007)
2. Azevedo, P.J., Jorge, A.M.: Comparing Rule Measures for Predictive Association Rules. In: Proceedings of ECML. pp. 510–517 (2007)
3. Chen, Y., Goldberg, S., Wang, D.Z., Johri, S.S.: Ontological Pathfinding: Mining First-Order Knowledge from Large Knowledge Bases. In: in Proc. of SIGMOD 2016. p. to appear (2016)
4. Galrraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast Rule Mining in Ontological Knowledge Bases with AMIE+. In: VLDB Journal (2015)
5. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proc. of 5th Int. Conf. and Symp. on Logic Programming, ICLP 1988. pp. 1070–1080 (1988)
6. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. 8(1), 53–87 (2004)
7. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The dlv system for knowledge representation and reasoning. ACM Transactions on Computational Logic (TOCL) 7(3), 499–562 (2006)
8. Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: A knowledge base from multilingual wikipedias. In: Proceedings of CIDR (2015)
9. Mohamed Gad-Elrab, Daria Stepanova, J.U.G.W.: Exception-enriched rule learning from knowledge graphs. In: Proceedings of the International Semantic Web Conference. TR available at `http://people.mpi-inf.mpg.de/~gadelrab/ExRules_TR.pdf` (2016)
10. Paes, A., Revoredo, K., Zaverucha, G., Costa, V.S.: Probabilistic first-order theory revision from examples. In: Inductive Logic Programming, 15th International Conference, ILP 2005, Bonn, Germany, August 10-13, 2005, Proceedings. pp. 295–311 (2005)
11. Ray, O.: Nonmonotonic abductive inductive learning. Journal of Applied Logic 3(7), 329–340 (2008)