Active Knowledge Graph Completion

Pouya Ghiasnezhad Omran[1,2], Kerry Taylor[1,3], Sergio Rodriguez Mendez[1,4],
and Armin Haller[1,5]

[1] Australian National University
[2] P.G.Omran@anu.edu.au
[3] kerry.taylor@anu.edu.au
[4] Sergio.RodriguezMendez@anu.edu.au
[5] armin.haller@anu.edu.au

Abstract. Knowledge graphs (KGs) proliferating on the Web are known to be incomplete. Much research has been proposed for automatic completion, sometimes by rule learning, that scales well. All existing methods learn closed rules. Here we introduce open path (OP) rules and present a novel algorithm, OPRL, for learning them. While closed rules are used to complete a KG by answering given queries, OP rules identify the incompleteness of a KG by inducing such queries to ask. We use adaptations of Freebase, YAGO2, and a synthetic but complete Poker KG to evaluate OPRL. We find that OPRL mines hundreds of accurate rules from massive KGs with up to 1M facts. The learnt OP rules induce queries with precision up to 98% and recall of 62% on a complete KG, demonstrating the first solution for *active* knowledge graph completion.

Keywords: Knowledge Graph Completion · Open Path Rule · Rule Learning · Knowledge Graph.

1 Introduction

Knowledge Graphs (KGs) are a convenient technology to model and store massive quantities of weakly-structured data. However, their intended scope is usually poorly defined and they fail to record relevant entities, as well as relevant relationships for the entities they do record. Techniques have been developed for knowledge graph completion and rule learning to curate KGs automatically [5]. In these approaches, models, often expressed as logical rules or vector embeddings, are learnt from a given KG. The models are then used for curating tasks including *link prediction* that predicts missing facts for extant entities.

Rule learning methods for KGs [7] consider *closed path* (CP) rules which are used to predict a ground fact that fully instantiates the triple at the head of the rule. Closed rules enable inference of specific facts that, if true, are missing from the KG. They draw attention to a potential missing fact only if the fully specified fact is able to be inferred by the rule. Thus KG completion is driven by the assumption that the KG *knows what it does not know*. In this paper we consider, for the first time, the problem of rules-based knowledge graph completion in the situation that the KG *does not know what it does not know*. This problem requires the KG, as it does for people, to look outside its own environment.

We propose learning *open path rules* (OP) from which we infer less constrained, open-ended queries to complete a knowledge graph. The proposed OP rule formalism is a fragment of existential rule [1] which is expressive enough to infer the queries yet they are suitable for our embedding-based scalable rule mining system. Our OP rules provide evidence that information is missing even when there is no evidence for a specific missing fact. The queries inferred from OP rules identify that a new fact is needed when the answer is not known, but also not obvious. Such queries could be posed to an active user engaged in a curating task or to a Web question-answering engine, where the answer might be found outside the KG. In particular, an answer to the query may well introduce previously unknown entities to the KG, and thus address a previously unstudied direction in knowledge graph completion, that is *missing entities*.

As a beneficial side-effect, our work addresses a long-standing gap in traditional link prediction systems (e.g. [2]), that use a KG to propose new links, but need to be seeded with queries about potential missing links. Such a query takes the form of a triple with one free variable. Conventionally, for evaluation purposes, these queries are generated from test facts that are held-back from the KG in the hope that a high-performing predictor will rediscover the held-back facts. However, once a link predictor is deployed over a working KG, test facts cannot be held back. Why would we want to remove a fact from the KG so that we can construct a query from it so that we might rediscover the same fact? And how should we choose which facts to remove to ensure that we are generating queries that are the most important to ask of our link predictor? Since generating queries this conventional way is problematic, whence *does* the query arise? We propose that the queries we infer from our OP rules can be widely used to generate the queries that link predictors need to repair KGs.

The contributions of this paper are as follows. We present a novel method for learning open path rules from a KG. These are Horn rules with a different form to the usual closed path rules that are used for knowledge graph completion tasks. We propose an algorithm to learn these rules based on the embedding presentations of the predicates and entities. As such, we introduce a first solution to the problem of *active knowledge graph completion* (AKGC), where we aim, instead of suggesting missing facts, to *ask the best questions* to complete a KG.

2 Learning Rules with Free Variables

Unlike earlier work in rule mining for KG completion, for our *active* knowledge graph completion task we mine *open path* (OP) rules of the following form:

$$P_t(x, z_0) \leftarrow P_1(z_0, z_1) \wedge P_2(z_1, z_2) \wedge ... \wedge P_n(z_{n-1}, y) \tag{1}$$

Here, P_i is a predicate in the KG and each of $\{x, z_i, y\}$ are variables (x and y are free while the z_is are bound). Unlike CP rules, OP rules do not necessarily form a looped path of variable connections, but every instantiation of a CP rule is also an instantiation of an OP rule. From an instantiation of the body of an OP rule, we can not infer a fact, but only a query. For example, the following OP rule, $citizenOf(x, t) \leftarrow livesIn(x, z)$. states that if an entity, x, lives in z, then that entity is citizen of somewhere (t). By instantiating the body of this rule as follows, $livesIn(bronte, canberra)$, we could infer the query, $citizenOf(bronte, ?)$.

To assess the quality of our mined open path rules, we introduce *open path standard confidence (OPSC)* and *open path head coverage (OPHC)* derived from the closed path forms.

Let r be an OP rule of the form (1). Then a pair of entities (e, e') *satisfies* the body of r, denoted $body_r(e, e')$, if there exist entities $e_1, ..., e_{n-1}$ in the KG such that $P_1(e, e_1)$, $P_2(e_1, e_2), ..., P_n(e_{n-1}, e')$ are facts in the KG. A pair (e', e) *satisfies* the head of r, denoted $P_t(e', e)$, if $P_t(e', e)$ is a fact in the KG. The *open path support*, *open path standard confidence*, and *open path head coverage* of r are given respectively by

$$OPsupp(r) = |\{e : \exists e', e'' \ s.t. \ body_r(e, e') \ and \ P_t(e'', e)\}|$$

$$OPSC(r) = \frac{OPsupp(r)}{|\{e : \exists e' \ s.t. \ body_r(e, e')\}|}, OPHC(r) = \frac{OPsupp(r)}{|\{e : \exists e' \ s.t. \ P_t(e', e)\}|}$$

OP Rule Learning: Our objective is to mine a KG for high-quality OP rules about a specific target predicate P_t. While we adhere to the architecture of RLvLR [7] that learns CP rules, we propose the following novelties for mining OP rules: (i) a novel fitness function which can estimate the quality of an OP rule based on the embedding representations of its predicates (based on the learnt embeddings from RESCAL [6]); and (ii) a novel vector computation which allows the system to evaluate the OP rules against a massive KG to compute quality measures, OPSC and OPHC.

3 Experiments

We have conducted two sets of experiments to evaluate oprl[6], demonstrating: (i) Oprl can mine quality OP rules from a range of KGs. Oprl can mine massive KGs in reasonable time. (ii) Queries generated from oprl's rules are relevant with good recall and precision in multiple KGs. They far outperform a distribution-based baseline. The four benchmark datasets are given in Table 1. All experiments were conducted on an Intel Xeon CPU E5-4620v2 @ 2.60GHz, 66GB RAM and running CentOS 7.

OP Rule Learning: First, we assess how well oprl finds high quality rules. We are not aware of other OP rule learners with which to compare, but we do compare the performance of fitness functions. The quality of rules are reported based on their OPSC/OPHC scores calculated against the full benchmark KGs,

[6] The datasets used in the experiments and detailed results can be found at www.dropbox.com/sh/y1f7zut09dheius/AADofv9c18Rzm-CFc64dw2yVa?dl=0

not the samples. Later we will use the mined rules for generating queries, so we need some holdout facts for query evaluation. For FB15KSE, test and training sets are available [7]. For Poker and YAGO2 core we can find no previously prepared data, so we randomly partition 90% for training and 10% for testing.

Table 1: Benchmark KG specifications

KG	# Facts	# Entities	# Predicates
FB15KSE [7]	272K	15K	237
YAGO2core [4]	948K	470K	32
Poker [3]	1M	95k	27

Table 2: Performance of OPRL on benchmark KGs

Benchmark	#Rules	#Arules	Time (hours)
FB15KSE	1029	261	0.17
YAGO2 core	84	9	0.20
Poker	603	509	0.52

Table 2 shows the average numbers of quality rules mined for all predicates and the running times (in hours, averaged over the targets). Similarly to [4], only rules with quality OPSC$\geq$ 0.1 and OPHC$\geq$ 0.01 are included. The average number of accurate rules, i.e. the rules with OPSC$\geq$ 0.8, are given as #Arules.

Query Generation: Our second set of experiments evaluates the predictive power of the mined rules satisfying quality thresholds OPSC$\geq$ 0.8 and OPHC$\geq$ 0.01 on the training data by posing the inferred queries to the test data. In the absence of any comparative system for query generation, we developed three baseline query sets (*Prand*) which contain random queries according to the frequency distribution of predicates and entities in the respective KG. Table 3 shows average precision, recall and F_1 where a query is considered true if it can be answered from the test data, and false otherwise. We see that OPRL's performance exceeds Prand on FB15KSE, YAGO2 core and Poker by factors of approximately 6, 2 and 9 respectively.

4 Conclusion

In this paper, we proposed a method for learning rules with free variables from Knowledge Graphs (KGs). Such rules can be used to generate queries soliciting missing facts. Notably, the queries could be fed to link predictors, so obtaining a fully automated framework for KG completion. Our experiments show that OPRL can learn rules for KGs with varying sizes and degrees of incompleteness. We show the usefulness of the mined rules by applying them to three different KGs to infer relevant queries.

Table 3: Accuracy of query generation

Benchmark	#Q	OPRL			Prand		
		P	R	F1	P	R	F1
FB15KSE	15k	**0.13**	**0.3**	**0.18**	0.02	0.05	0.03
YAGO2 core	9k	**0.14**	**0.01**	**0.03**	0.06	0.005	0.01
Poker	41k	**0.98**	**0.62**	**0.76**	0.17	0.07	0.1

References

1. Bellomarini, L., Sallinger, E., Gottlob, G.: The Vadalog system: Datalogbased reasoning for knowledge graphs. In: VLDB. vol. 11, pp. 975–987 (2018)
2. Bordes, A., Usunier, N., Weston, J., Yakhnenko, O., Garcia-Duran, A.: Translating embeddings for modeling multi-relational data. In: NIPS. pp. 2787–2795 (2013)
3. Dua, D., Graff, C.: UCI machine learning repository (2017), archive.ics.uci.edu/ml Retrieved Nov 2019
4. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with AMIE+. The VLDB Journal pp. 707–730 (2015)
5. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A Review of Relational Machine Learning for Knowledge Graphs. In: IEEE. vol. 104, pp. 11–33 (2016)
6. Nickel, M., Rosasco, L., Poggio, T.: Holographic Embeddings of Knowledge Graphs. In: AAAI. pp. 1955–1961 (2016)
7. Omran, P.G., Wang, K., Wang, Z.: Scalable Rule Learning via Learning Representation. In: IJCAI. pp. 2149–2155 (2018)