A System for Reasoning-based Link Prediction in Large Knowledge Graphs

Hong Wu[1], Zhe Wang[2], Xiaowang Zhang[1], Pouya Ghiasnezhad Omran[3], Zhiyong Feng[1] and Kewen Wang[2⋆]

[1] College of Intelligence and Computing, Tianjin University, China
[2] School of Information and Communication Technology, Griffith University, Australian
[3] Research School of Computer Science, Australian National University, Australia

Abstract. This poster paper presents an efficient method R-Linker for link prediction in large knowledge graphs, based on rule learning. The scalability and efficiency is achieved by a combination of several optimisation techniques. Experimental results show that R-Linker is able to handle KGs with over 10 million of entities and more efficient than existing state-of-the-art methods including RLvLR and AMIE+ in rule learning stage for link prediction.

1 Introduction

Knowledge graphs (KGs), a new generation of knowledge bases, have received significant attention in semantic technologies. As a KG is usually very large (of size over 10 million entities), it is infeasible for manual construction. Also, KGs are usually incomplete. Thus, it is useful and challenging to automatically construct and enrich KGs. Link prediction is one of important tasks for automated construction of KGs. Given an entity e and a (binary) relation R, the problem of link prediction is to find an entity e' such that the triple (e, R, e') (or equivalently, the fact $R(e, e')$) is in the KG. A large number of methods for link prediction have bee proposed in the literature, including Neural LP, Node+LinkFeat and DISMULT [1]. However, most of these methods work only for relatively small KGs like WN18 and FB15K.

AMIE+ [4] and RLvLR [6] are among more recent methods that are able to predict links for larger KGs of size over 10 millions, and thus these methods are much more scalable than other rule learners such as [3,5]. As these two methods are essentially rule learners, they can address the link prediction in a more general form. For convenience, we refer this link prediction as *Reasoning-based Link Prediction* or just *R-link prediction*. Specifically, given a relation R, we want to find a pair (or pairs) e and e' of entities such that triple (e, R, e') is in the KG. In particular, RLvLR demonstrates that the technique of embedding in representation learning is promising for handling R-link prediction problem in large KGs. In order to extract information on the nationality of persons, one can learn a rule like $\mathsf{BornIn}(x, y) \wedge \mathsf{Country}(y, z) \rightarrow \mathsf{Nationality}(x, z)$. Rules are explicit knowledge (compared to a neural network) and can provide human understandable explanations to learning results (e.g., link prediction) based on them. Thus, it is useful and important to extract rules for KGs automatically.

⋆ The corresponding author: k.wang@griffith.edu.au

In this poster paper, we further push the envelope by developing a more efficient method R-Linker for R-link prediction in large KGs. The scalability and efficiency of R-Linker is achieved by a combination of several optimisation techniques. First, we use an adapted embedding for rule learning; Second, we introduce a new strategy of sampling called *Hierarchical Sampling*; Moreover, we develop new techniques of rule search and rule evaluation. As a result, we have implemented a new system R-Linker for link prediction with large KGs. Our experiments show that R-Linker is able to handle KGs of size over 10M and more efficient than other methods including RLvLR and AMIE+ in rule learning stage for link prediction. R-Linker is available at `https://www.dropbox.com/sh/c8ent25u3qp4vp1/AABc6Jl3zTRtOkdTwHaoDBDUa?dl=0`

2 A Rule-based Model

Unlike other statistical relational models, we adopt rule-based models for link prediction, with the obvious advantage that the learned models (as sets of logical rules) are explainable and reusable. In what follows, we describe how we construct such models.

2.1 Embedding-based Rule Selection

Inspired by [6], we learn such rule-based models via predicate embeddings; yet unlike [6] using matrix embeddings, we adopt TransE vector embeddings which can significantly improve learning efficiency. As we demonstrate in the experiments, adopt a simpler form of embeddings does not compromise the learning accuracy. In [2], vector embeddings $\boldsymbol{r}$ and $\boldsymbol{e}$ are constructed for each relation R and each entity e in the KG. When a fact $R(e, e')$ exists in the KG, the embeddings satisfy $\boldsymbol{e}+\boldsymbol{r} \approx \boldsymbol{e}'$. We extend it to an embedding characterisation for closed-path rules, that is, first-order Horn rules of the form $R_1(x, z_1) \wedge R_2(z_1, z_2) \wedge \ldots \wedge R_n(z_{n-1}, y) \rightarrow R(x, y)$ with $x, y, z_1, \ldots, z_{n-1}$ being variables. There are two aspects we hope to capture: (1) the composition of relations $R_1, \ldots R_n$ associates entities (in place of x and y) similarly as relation R does; and (2) the co-occurrence of arguments in the positions of $x, y, z_1, \ldots, z_{n-1}$.

For (1), it requires for each pair of entities (e, e'), $\boldsymbol{e}+\boldsymbol{r}_1+\cdots+\boldsymbol{r}_n-\boldsymbol{e}' \approx \boldsymbol{e}+\boldsymbol{r}-\boldsymbol{e}'$. We define a measure $sim(\boldsymbol{r}_1+\cdots+\boldsymbol{r}_n, \boldsymbol{r})$, where sim is the L2 norm of vector distances. For (2), we use the notion of argumentation embeddings from [6]. More specifically, for each relation R, two vector embeddings $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$ are computed by averaging the entity embeddings (as vectors) of all the entities occurring in the position of respectively, the subject and object arguments of R. Then, for x occurring as the subject arguments of both R_1 and R, y occurring as the object argument of both R_n and R, and z_i $(1 \leq i \leq n-1)$ occurring as the object argument of R_i and subject argument of R_{i+1}, we have the following measure $sim(\boldsymbol{r}_1^1, \boldsymbol{r}^1)+sim(\boldsymbol{r}_n^2, \boldsymbol{r}^2)+sim(\boldsymbol{r}_1^2, \boldsymbol{r}_2^1)+\cdots+sim(\boldsymbol{r}_{n-1}^2, \boldsymbol{r}_n^1)$.

2.2 Hierarchical Data Sampling

A major challenge in the computation of embeddings is that existing methods cannot scale over large KGs, even for vector embeddings. Hence, we propose a new data sampling strategy, called hierarchical sampling, to reduce the sizes of input KGs by

focusing on entities that are relevant to the link prediction task. Intuitively, for each link prediction task, the link (i.e., a relation) R is often given, and we sample entities (and facts) in the KG that are directly or indirectly related to R for embedding construction.

Consider a KG $K = (E, F)$ with E being the set of all entities and F being the set of all facts (i.e., triples) in K. Our sampling method selects a (small) subset $E' \subseteq E$ that are relevant to R and focus on the facts F' only about E' (not mentioning other entities). Since each rule in our model forms a path, our sampling method also deploys a breath-first tree search. As shown in Figure 1 (a), the first sampled entities E_0 are those occurring in facts about R. Then, E_1 are those entities that occur in any facts (not necessarily about R) mentioning entities from E_0. Similarly, E_{i+1} are those entities that occur in any facts mentioning entities from E_i, for each $i \geq 1$ till a prescribed depth.

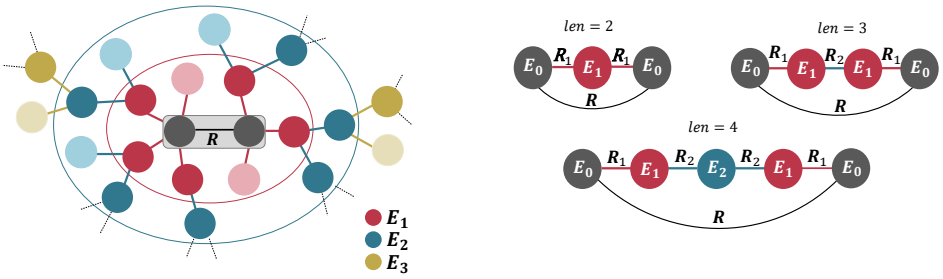

Fig. 1: (a) Breath-first search for sampling; (b) Sampling preserves closed-paths.

Figure 1 (b) shows how our sampling method preserves closed-path rules. For a rule of length 3, $R_1(x, z) \wedge R_2(z, y) \rightarrow R(x, y)$ and each supporting instance of the rule $R_1(e, e'') \wedge R_2(e'', e') \rightarrow R(e, e')$, entities e and e' will be sampled in E_0 and e'' in E_1.

One optimisation is loop elimination during the breath-first search, as shown in Figure 1 (a), if a repeated entity is found on a path (represented in light color), the path is no longer explored. This is to avoid redundant atoms in the rules, for example $R_1(x, y) \wedge R_1^-(y, x) \wedge R_1(x, y) \rightarrow R(x, y)$. Furthermore, by recording the path information during the search, it eliminates a large number of invalid compositions of relations and can effectively suggest candidate rules. Other optimisations include selecting a bounded number of neighbours for each entity, and pruning relations with low frequency.

The evaluation of candidate rules, through the computation of standard confidence and head coverage, is often expensive, and much research effort has be dedicated to optimise such computation. A key step is to compute the support degree, i.e., the number of entity pairs in KG that make both body and head of the rule true. From the above discussions, we can quickly narrow down our search to entities directly connected to those in E_0, and since the relation in the head is known, we can first check whether a pair of entities satisfy the head. These optimisations prove to be quite effective.

3 Experiments

We compared our system with RLvLR, AMIE+ and Neural LP on rule learning and link prediction, on common benchmarks FB15K(-237), Wikidata, DBPedia 3.8, and YAGO2s.

For large KGs Wikidata, DBPedia 3.8, and YAGO2s, Table 1 shows our system outperforms both RLvLR and AMIE+ in learning efficiency, as shown by the average numbers of rules (#R) and quality rules (#QR, standard confidence over 0.7) learned per hour.

Table 1: Rule learning on large KGs.

KG	R-Linker		RLvLR		AMIE+	
	#R	#QR	#R	#QR	#R	#QR
DBpedia	**13.38**	**3.67**	11	2.37	1.97	0.11
Wikidata	**37.38**	**18.52**	23.56	10.62	<0.09	<0.03
YAGO2s	**9.71**	**2.28**	6.56	1.88	<0.56	<0.05

Table 2 shows that compared to RLvLR and Neural LP, the model constructed by our system demonstrates better accuracy on link prediction. Table 2 shows the comparison of our rule-based model against statistical models on FB15K-237. While our model has competitive performance on link prediction, its major advantage is that rule-based models are explainable and reusable. We plan to compare our method with some other approaches such as [7].

Table 2: Link prediction on large KGs.

Learner	FB75K		Wikidata	
	MRR	Hits@10	MRR	Hits@10
R-Linker	**0.37**	**59.0**	**0.33**	**39.3**
RLvLR	0.34	43.4	0.29	38.9
Neural LP	0.13	25.7	-	-

Table 3: Link prediction on FB15K-237.

Learner	MRR	Hits@10
DISTMULT	0.25	40.8
Node+LinkFeat	0.23	34.7
Neural LP	0.24	36.1
RLvLR	0.24	39.3
R-Linker	0.24	38.1

References

1. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A Review of Relational Machine Learning for Knowledge Graph. Proceedings of IEEE, 1041, 1-23 (2016).
2. Antoine, B., Nicolas, U., Alberto, G.D., Jason, W., Oksana, Y.: Translating embeddings for modeling multi-relational data. In: NIPS 26, pp. 2787–2795 (2013).
3. Chen, Y., Wang, D.Z., Goldberg, S.: Scalekb: scalable learning and inference over large knowledge bases. The VLDB Journal **25**(6), 893–918 (2016).
4. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with amie+. The VLDB Journal **24**(6), 707–730 (2015).
5. Ho, V.T., Stepanova, D., Gad-Elrab, M.H., Kharlamov, E., Weikum, G.: Rule learning from knowledge graphs guided by embedding models. In Proc. ISWC pp. 72–90 (2018)
6. Omran, P.G., Wang, K., Wang, Z.: Scalable rule learning via learning representation. In Proc. AAAI, pp. 2149–2155 (2018).
7. García-Durán, A., Niepert, M.: Blrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features. In: Proc. UAI, pp, 372-381 (2018).