UniKER: A Unified Framework for Combining Embedding and Horn Rules for Knowledge Graph Inference

Kewei Cheng [1] **Ziqing Yang** [2] **Ming Zhang** [2] **Yizhou Sun** [1]

Abstract

Combining KGE and logical rules for better KG inference has gained increasing attention in recent years. Unfortunately, a majority of existing methods employ sampling strategies to randomly select only a small portion of ground rules or hidden triples, thus can only partially leverage the power of logical rules in reasoning. In this paper, we propose a novel framework UniKER to address this challenge by restricting logical rules to be Horn rules, which can fully exploit the knowledge in logical rules and enable the mutual enhancement of logical rule-based reasoning and KGE in an extremely efficient way. Extensive experiments have demonstrated that our approach is superior to existing state-of-the-art algorithms in terms of both efficiency and effectiveness.

1. Introduction

Knowledge graph inference has been studied extensively due to its wide applications in different domains, such as search engines and question answering systems. There are two main directions in solving the inference problem, i.e., logical rule reasoning and knowledge graph embedding (KGE). Both methods have their own superiority as well as limitations. On one hand, although logical rule-based approaches have shown their strong ability to capture high-order dependency between entities and relations, they suffer from incapability to handle noisy data due to their symbolic nature. In addition, high computation complexity presents another central challenge for logical rule-based approaches. On the other hand, even though KGE methods have demonstrated their good scalability when coping with large scale real-world KGs, they fail to capture high-order dependency between entities and relations.

Since KGE methods and logical rule-based methods are complementary for better reasoning capability, several attempts have been made to combine KGE and logical rules for better KG inference. However, most of them (Guo et al., 2016; Rocktäschel et al., 2015; Demeester et al., 2016) only make a one-time injection of logic rules to KG embeddings and thus fail to capture the mutual interaction between KGE and logical rules (Guo et al., 2016; Rocktäschel et al., 2015). Also, all the existing methods model logical inference as an NP-complete problem by ignoring the fact that only Horn rules, a special type of logical rules, are used for most time in reality. As a result, to improve the scalability of logical inference, they use sampling strategies that select only a small portion of hidden triples/ground rules to approximate the inference process, which inevitably causes loss of information from the logical side. To address the above issues, we propose a novel framework, UniKER, to combine KGE and logical rules for better KG inference in an iterative manner. In particular, by leveraging the nice properties of Horn rules, UniKER can fully exploit the knowledge contained in logical rules and completely transfer them into the embeddings. Additionally, UniKER can also tolerate erroneous data and show robustness to noise and error in the KGs, which previous methods cannot cope with.

2. Preliminaries and Related Work

Knowledge Graphs in the Language of Symbolic Logic. A knowledge graph, denoted by $\mathcal{G} = \{E, R, O\}$, consists of a set of entities E, a set of relations R, and a set of observed facts O. Each fact in O is represented by a triple (e_i, r_k, e_j), where $e_i \in E$, $e_j \in E$, and $r_k \in R$ denote subject entity, object entity, and relation, respectively. In the area of symbolic reasoning, entities can also be considered as **constants** and relations are called **predicates**. Each predicate in KGs is a binary logical function defined over two constants, denoted as $r(\cdot, \cdot)$. **A ground predicate** is a predicate whose arguments are all instantiated by particular constants. For example, we may have a predicate $\text{Friend}(\cdot, \cdot)$. By assigning constants Alice and Bob to it, we get a ground predicate $\text{Friend}(\text{Alice}, \text{Bob})$. A

[1]Department of Computer Science, University of California, Los Angeles [2]Department of Computer Science, Peking University. Correspondence to: Kewei Cheng, Yizhou Sun <{viviancheng,yzsun}@cs.ucla.edu>, Ziqing Yang <yangziqing@pku.edu.cn>, Ming Zhang <mzhang@net.pku.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

triple (e_i, r_k, e_j) is essentially a ground predicate, denoted as $r_k(e_i, e_j)$ in the language of logic. In the reasoning task, a ground predicate can be regarded as a binary random variable: $r_k(e_i, e_j) = 1$ when the triple (e_i, r_k, e_j) holds true, and $r_k(e_i, e_j) = 0$ otherwise. Given the observed facts and their corresponding ground predicates $\mathbf{v}_O = \{r_k(e_i, e_j)|(e_i, r_k, e_j) \in O\}$, the task of **knowledge graph inference** is to predict the truth value of ground predicates $\mathbf{v}_H = \{r_k(e_i, e_j)|(e_i, r_k, e_j) \in H\}$ for all remaining hidden triples, where $H = \{(e_i, r_k, e_j)|(e_i, r_k, e_j) \notin O, e_i, e_j \in E, r_k \in R\}$.

First Order Logic and Horn Rules. First-order logic (FOL) rules are constructed over predicates using logical connectives and quantifiers. They usually require extensive human supervision to create and validate, which severely limit their applications. Instead, **Horn rules**, as a special case of FOL rules, can be extracted automatically and efficiently via modern rule mining systems, such as WARMR (Dehaspe & Toivonen, 1999) and AMIE (Galárraga et al., 2013; 2015) with high quality, thus result in their monopoly in practice. Horn rules are composed a body of conjunctive predicates and a single head predicate. They are usually written in the form of implication and an example is shown below:

$$\forall x, y, z : r_0(x, y) \leftarrow r_1(x, z_1) \wedge r_2(z_1, z_2) \wedge r_3(z_2, y) \quad (1)$$

where $r_0(x, y)$ is called the **head** of the rule while $r_1(x, z_1) \wedge r_2(z_1, z_2) \wedge r_3(z_2, y)$ is the **body** of the rule. By substituting the variables x, z_1, z_2, y with concrete entities e_i, e_p, e_q, e_j, we get a ground Horn rule as follows:

$$r_0(e_i, e_j) \leftarrow r_1(e_i, e_p) \wedge r_2(e_p, e_q) \wedge r_3(e_q, e_j) \quad (2)$$

A Brief Review over Knowledge Graph Inference There are two main directions in solving the KG inference problem, i.e., traditional logical inference and KGE. Traditional logical inference aims to find an assignment of truth values to all hidden ground predicates, which results in maximizing the number of ground rules that can be satisfied. Thus, it can be mathematically modeled as a MAX-SAT problem, which is NP-complete (Arora & Barak, 2009). One approach to this problem is stochastic local search, exemplified by WalkSAT (Selman et al., 1993). Markov Logic Network (MLN) (Richardson & Domingos, 2006) further provides a probabilistic extension of FOL via probabilistic graphical models. Unlike traditional logical inference, which infer missing facts via logical rules, KGE aims to capture the similarity of entities by embedding entities and relations into continuous low-dimensional vectors. Scoring functions (SFs), which measure the plausibility of triples in KGs, is the crux of KGE models. We denote the score of a triple (e_i, r_k, e_j) calculated following SF as $f_{r_k}(e_i, e_j)$. Representative KGE algorithms include TransE (Bordes et al., 2013), TransH (Wang et al., 2014), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016) and RotatE (Sun et al., 2019), which differ from each other with different SFs.

Several attempts have been made to combine KG embedding and logical rules for better KG inference, which can be broadly divided into two categories: (1) designing logical rule-based regularization to embedding models. Approaches in this category treat logical rules as additional regularization to embedding models, where the satisfaction loss of ground rules is integrated into the original embedding loss. The satisfication loss of a ground rule is usually computed based on soft logic, where the probability of each predicate is determined by the embedding. KALE (Guo et al., 2016), RUGE (Guo et al., 2017) and Rocktäschel et al. (Rocktäschel et al., 2015) are some of the representative methods; and (2) designing embedding-based variational distribution for variational inference of MLN. Several methods including pGAT (Harsha Vardhan et al., 2020), ExpressGNN (Zhang et al., 2019) and pLogicNet (Qu & Tang, 2019) propose to leverage graph embedding to define variational distribution for all possible hidden triples to conduct variational inference of MLN.

3. A Unified Framework for Knowledge Graph Inference: UniKER

Both types of existing approaches consider logical rule inference as an NP complete problem by ignoring the fact that in most cases only Horn rules, a special case of logical rules, are used in reality. Due to the complexity of NP complete problems, these methods only partially leverage the power of logical rules in reasoning by sampling a small portion of hidden triples/ground rules to avoid infeasible inference time. In this section, we show that by leveraging the nice properties of Horn rules, there is a much simpler way to directly derive truth values of all unobserved triples.

Horn-satisfiability of Knowledge Graph Inference Given a set of Horn rules $\mathcal{F}$ and their ground Horn rules $\mathcal{F}_g$, if there exists at least one truth assignment that satisfies all ground Horn rules $\mathcal{F}_g$, we call it *Horn-satisfiable*. We will show there always exists a truth assignment to all hidden triples in a KG such that all ground Horn rules are satisfied, i.e., Horn-satisfiable.

Theorem 1. *Knowledge graph inference is Horn-satisfiable.*

Proof. A set of ground Horn rules is unsatisfiable if we can derive a pair of opposite ground predicates (i.e., $r_0(e_i, e_j)$ and $\neg r_0(e_i, e_j)$) from them. It is the case if and only if $\neg r_0(e_i, e_j)$ is defined in KG as Horn rules can only include one single positive head predicate which results in its incapability in deriving negative triples. However, a typical KG will not explicitly include negative triples (i.e., $\neg r_0(e_i, e_j)$). Thus we can never derive such a pair of opposite ground predicates, which confirms that KG inference is Horn-satisfiable. $\square$

Truth Value Assignment via Forward Chaining According to Theorem 1, it is guaranteed that there exists a truth assignment that satisfies all ground Horn rules, which can be denoted as $\mathbf{v}_H^{T^*}$ and $\mathbf{v}_H^{F^*}$, where $\mathbf{v}_H^{T^*} = \{r_k(e_i, e_j) = 1 \mid r_k(e_i, e_j) \in \mathbf{v}_H\}$ and $\mathbf{v}_H^{F^*} = \{r_k(e_i, e_j) = 0 \mid r_k(e_i, e_j) \in \mathbf{v}_H\}$. An existing algorithm called **forward chaining** (Salvat & Mugnier, 1996) has been proposed to derive $\mathbf{v}_H^{T^*}$ and $\mathbf{v}_H^{F^*}$ in an efficient way. The basic mechanism is that starting from any ground rule whose bodies are satisfied in the KG, it keeps adding the inferred head (i.e., the new knowledge represented by a ground predicate) to the KG until no ground predicate can be added anymore. Unlike other logical inference algorithms, which require all ground rules into calculation, forward chaining adopts lazy inference instead. It actives only ground rules whose bodies are satisfied in the KGs to add the inferred head (i.e., the new knowledge represented by a ground predicate) to the KGs until no more head predicate can be inferred. The mechanism dramatically improves inference efficiency via avoiding the computation for a large number of ground predicates/rules that are never used.

Enhancement of Logical Inference via Knowledge Graph Embedding Although forward chaining can find the satisfying truth assignment for all hidden triples in an efficient way, its reasoning ability is severely limited by the coverage of rules, the incompleteness of the KG, and the errors contained in KG. Fortunately, due to its strong reasoning ability and robustness, KGE models are not only useful to prepare a more complete KGs by including useful hidden triples but also helpful to eliminate incorrect triples in both KGs and inferred results.

Including Potential Useful Hidden Triples. Due to the sparsity of real-world KGs, only a small portion of ground Horn rules can contribute to logical inference, as a ground Horn rule can get activated only if all the predicates in its body are completely observed, which severely limits the reasoning ability of Horn rules. A straightforward solution would be computing the score for every hidden triple and adding the most promising ones with the highest scores to the KG. Unfortunately, the number of hidden triples is quadratic to the number of entities (i.e. $O(|E| \times |R| \times |E|)$), thus it is too expensive to compute scores for all of them. Instead, we adopt "lazy inference" strategy to select only a small subset of "potential useful" triples. To illustrate what is a "potential useful" triple, we take the ground Horn rule in Eq. (2) as an example. If $r_1(e_i, e_p) \in v_O$, $r_3(e_q, e_j) \in v_O$, and $r_2(e_p, e_q) \in v_H$, we would not be able to infer the head (i.e., $r_0(e_i, e_j)$) as whether $r_2(e_p, e_q)$ is true or not is unknown. Thus, $r_2(e_p, e_q)$ becomes the crux to determine the truth value of the head, which is called "potential useful". In general, given a ground rule whose body includes only one unobserved ground predicate, this unobserved ground predicate can be regarded as a "potential useful" triple. We denote the set of all 'potential useful" triples as Δ_+. The detailed algorithm of identifying 'potential useful" triples can be found in appendix.

Excluding Potential Incorrect Triples. In addition, due to the symbolic nature, logical rules also lack the ability to handle noisy data. If the KGs contain any error, based on incorrect observations, forward chaining will not be able to make the correct inference. Even worse, it might contribute to the propagation of the error by including incorrectly inferred triples. Therefore, eliminating incorrect triples in both KGs and inferred results is significant for logical inference. Since KGE models show great power in capturing network structure of KGs, which incorrect triples usually contradict, error triples usually get lower prediction scores in KGE models compared to correct ones. For each triple (e_i, r_k, e_j) in $O \cup \mathbf{V}_H^{T^*}$, score $f_{r_k}(e_i, e_j)$ will be computed by KGE model to measure its reliability. We denote bottom $\theta\%$ triples with lowest prediction scores as Δ_-. It will be excluded from $O \cup \mathbf{V}_H^{T^*}$ to alleviate the impact of noise.

Enhancement of Knowledge Graph Embedding via Logical Inference Since $\mathbf{v}_H^{T^*}$ and $\mathbf{v}_H^{F^*}$ are the satisfying truth assignment derived by forward chaining, knowledge contained in Horn rules is guaranteed to be fully exploited by taking $\mathbf{v}_H^{T^*}$ and $\mathbf{v}_H^{F^*}$ as guidance to to optimize KGE model. Thus, the objective function of KGE model becomes as follows:

$$\begin{aligned}\min_{\{\mathbf{e}\},\{\mathbf{r}\}} & \sum_{(e_i,r_k,e_j)\in(O\cup\mathbf{v}_H^{T^*})} \max(0, \gamma - f_{r_k}(e_i, e_j) \\ & + \sum_{(e'_i,r,e'_j)\in\mathcal{N}(e_i,r,e_j)} \frac{1}{|\mathcal{N}(e_i, r, e_j)|} f_{r_k}(e'_i, e'_j))\end{aligned} \tag{3}$$

where a common margin-based pairwise ranking loss is employed to define the objective function. When learning the entity and relation embeddings, we treat triples (e_i, r_k, e_j) in both O and $\mathbf{v}_H^{T^*}$ as positive examples while (e'_i, r_k, e'_j) is their corresponding negative samples, and γ is a margin separating them. The score $f_{r_k}(e_i, e_j)$ of a triple (e_i, r_k, e_j) can be calculated following any SFs of KGE models. To reduce the effects of randomness, we sample multiple negative triples for each positive sample. We denote the negative triple set of a positive triple (e_i, r_k, e_j) as $\mathcal{N}(e_i, r_k, e_j)$. Conventional embedding models follow closed world assumption (CWA) (i.e., assuming all facts

Model	Kinship Hit@1	Kinship Hit@10	Kinship MRR	FB15k-237 Hit@1	FB15k-237 Hit@10	FB15k-237 MRR	WN18RR Hit@1	WN18RR Hit@10	WN18RR MRR
RESCAL (Nickel et al., 2011)	0.489	0.894	0.639	0.108	0.322	0.179	0.123	0.239	0.162
SimplE (Kazemi & Poole, 2018)	0.335	0.888	0.528	0.150	0.443	0.249	0.290	0.351	0.311
KALE (Guo et al., 2016)	0.433	0.869	0.598	0.131	0.424	0.230	0.032	0.353	0.172
RUGE (Guo et al., 2017)	0.495	0.962	0.677	0.098	0.376	0.191	0.251	0.327	0.280
BLP (De Raedt & Kersting, 2008)†	-	-	-	0.062	0.150	0.092	0.187	0.358	0.254
MLN (Richardson & Domingos, 2006)†	0.655	0.732	0.694	0.067	0.160	0.098	0.191	0.361	0.259
ExpressGNN (Zhang et al., 2019)	0.105	0.282	0.164	0.150	0.317	0.207	0.036	0.093	0.054
pLogicNet (Qu & Tang, 2019)†	0.683	0.874	0.768	0.237	0.524	0.332	0.398	0.537	0.441
pGAT (Harsha Vardhan et al., 2020)‡	-	-	-	0.377	0.609	0.457	0.395	**0.578**	0.459
TransE (Bordes et al., 2013)†	0.221	0.874	0.453	0.198	0.441	0.279	0.013	0.531	0.223
UniKER-TransE	**0.866**	**0.968**	**0.910**	0.463	**0.630**	0.522	0.040	0.561	0.307
DistMult (Toutanova et al., 2015)†	0.360	0.885	0.543	0.199	0.446	0.281	0.390	0.490	0.430
UniKER-DistMult	0.770	0.945	0.823	**0.507**	0.587	**0.533**	**0.432**	0.538	**0.485**

† Results on FB15k-237 and WN18RR are taken from (Qu & Tang, 2019).
‡ Results are taken from (Harsha Vardhan et al., 2020).

Table 1. Results of Reasoning on Kinship, FB15K-237 and WN18RR Datasets.

that are not contained in the knowledge graph are false) to construct negative triples, which is usually incorrect in real-world applications. Instead of adopting CWA, we conduct negative sampling from $\mathbf{v}_H^{F*}$ to make sure that true but unseen triples will not be sampled. As assignment $\mathbf{v}_H^{T*}$ and $\mathbf{v}_H^{F*}$ is a satisfying truth assignment regard to the HORN-SAT problem defined over all ground Horn rules, we can safely regard any hidden triples which belong to $\mathbf{v}_H^{F*}$ as the negative triples without violating any ground Horn rules.

Integrating Embedding and Logical Rules in an Iterative Manner. Since logical rules and KGE can mutually enhance each other as discussed above, we propose a unified framework, known as UniKER, to integrate KGE and Horn rules-based inference in an iterative manner. For each iteration, it is comprised of two steps. First, following forward chaining algorithm, we derive entailed triples set $\mathbf{v}_H^{T\,i*}$ based on current KG (i.e., O). Then, we add newly inferred triples $\mathbf{v}_H^{T\,i*}$ to KG by updating $O = O \cup \mathbf{v}_H^{T\,i*}$. Second, we train a KGE model based on the updated KG (i.e., O). With the well trained KGE, we eliminate Δ_-, which is the bottom $\theta\%$ triples with lowest prediction scores, from O meanwhile add new potentially useful triples Δ_+ to O.

4. Experiments

Knowledge Graph Completion We compare different algorithms on KG inference task. We mask the head or tail entity of each test triple, and require each method to predict the masked entity. During evaluation, we use the filtered setting (Bordes et al., 2013) and three evaluation metrics, i.e., Hit@1, Hit@10 and Mean Reciprocal Rank (MRR). Table 1 shows the comparison results from which we find that: (1) UniKER consistently outperforms basic KGE models in almost all cases with significant performance gain, which can ascribe to the utilization of additional knowledge from logical rules; (2) UniKER also obtains better performance than both classes of approaches to combine embedding model with logical rules as it provides an exact optimal solution to HORN-SAT problem defined over all ground Horn rules rather than employ sampling strategies to do approximation; (3) Traditional rule-based algorithms show the worst performance among all methods. The major reason is the insufficient coverage of logical rules, which indicates the potential of using KGE to improve the reasoning ability of traditional rule-based algorithms.

Impact of Iterative Algorithm on KG Completion. Note that UniKER is trained in an iterative way. In each iteration, there are some new triples being derived. To investigate how this iterative process helps improve reasoning ability of UniKER, we conduct experiments on Kinship dataset. In particular, iteration 0 represents KGE model is trained based on the original data without any inferred triples included. As presented in Figure 1, we observed that (1) With the increase of iterations, the performance is first improved rapidly, then slows down gradually; (2) UniKER has a bigger impact on Hit@1, Hit@10 compared to MRR.

Robustness Analysis. To investigate the robustness of UniKER, we compare the reasoning ability of UniKER with TransE on Kinship dataset with noise. We introduce noise by substituting the true head entity or tail entity with randomly selected entity. Following this approach, we construct a noisy Kinship dataset with noisy triples to be 40% of original training data. To study the effect of parameter θ (i.e., the threshold used to eliminate noisy triples), we vary θ among $\{10, 20, 30\}$. The comparison results are presented in Table 2. We can observe that (1) UniKER outperforms TransE on noisy KG with significant performance gain; (2) With the increase of θ, the performance of UniKER keeps improving, which validates that our UniKER can indeed eliminate noise from training data.

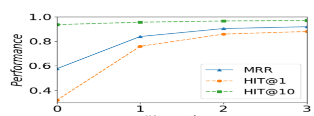

Figure 1. Performance of KG Completion on Kinship Dataset w.r.t. #Iterations for Effectiveness Analysis.

Model	θ	Hit@1	Hit@10	MRR
TransE	-	0.026	0.800	0.319
UniKER-TransE	10	0.286	0.776	0.466
UniKER-TransE	20	0.311	0.816	0.503
UniKER-TransE	30	**0.322**	**0.833**	**0.520**

Table 2. Results of Reasoning on Kinship Dataset with Noise. $\theta\%$ is the Threshold Used to Eliminate Noise.

5. Conclusion

In this paper, we proposed a novel framework, known as UniKER, to integrate embedding and Horn rules in an iterative manner for better KG inference. We have shown that UniKER can fully leverage the knowledge from Horn rules and completely transfer them into the embedding models in an extremely efficient way. In addition, UniKER also shows robustness to noise and error in KGs, which previous methods cannot cope with. The experimental results demonstrate that UniKER is superior to existing state-of-the-art algorithms in terms of both efficiency and effectiveness.

Acknowledgments

Ziqing Yang and Ming Zhang are supported by National Key Research and Development Program of China with Grant No. 2018AAA0101900 / 2018AAA0101902 as well as the National Natural Science Foundation of China (NSFC Grant No. 61772039 and No. 91646202).

References

Arora, S. and Barak, B. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pp. 2787–2795, 2013.

De Raedt, L. and Kersting, K. Probabilistic inductive logic programming. In *Probabilistic Inductive Logic Programming*, pp. 1–27. Springer, 2008.

Dehaspe, L. and Toivonen, H. Discovery of frequent datalog patterns. *Data Mining and knowledge discovery*, 3(1): 7–36, 1999.

Demeester, T., Rocktäschel, T., and Riedel, S. Lifted rule injection for relation embeddings. *arXiv preprint arXiv:1606.08359*, 2016.

Denham, W. W. *The detection of patterns in Alyawara nonverbal behavior*. PhD thesis, University of Washington, Seattle., 1973.

Galárraga, L., Teflioudi, C., Hose, K., and Suchanek, F. M. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal—The International Journal on Very Large Data Bases*, 24(6):707–730, 2015.

Galárraga, L. A., Teflioudi, C., Hose, K., and Suchanek, F. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 413–422. ACM, 2013.

Guo, S., Wang, Q., Wang, L., Wang, B., and Guo, L. Jointly embedding knowledge graphs and logical rules. pp. 192–202, 01 2016. doi: 10.18653/v1/D16-1019.

Guo, S., Wang, Q., Wang, L., Wang, B., and Guo, L. Knowledge graph embedding with iterative guidance from soft rules, 2017.

Harsha Vardhan, L. V., Jia, G., and Kok, S. Probabilistic logic graph attention networks for reasoning. In *Companion Proceedings of the Web Conference 2020*, WWW '20, pp. 669–673, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370240. doi: 10.1145/3366424.3391265. URL https://doi.org/10.1145/3366424.3391265.

Kazemi, S. M. and Poole, D. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*, pp. 4284–4295, 2018.

Nickel, M., Tresp, V., and Kriegel, H.-P. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pp. 809–816, 2011.

Qu, M. and Tang, J. Probabilistic logic neural networks for reasoning, 2019.

Richardson, M. and Domingos, P. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

Rocktäschel, T., Singh, S., and Riedel, S. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1119–1129, 2015.

Salvat, E. and Mugnier, M.-L. Sound and complete forward and backward chainings of graph rules. In *International Conference on Conceptual Structures*, pp. 248–262. Springer, 1996.

Selman, B., Kautz, H. A., Cohen, B., et al. Local search strategies for satisfiability testing. *Cliques, coloring, and satisfiability*, 26:521–532, 1993.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, 2015.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pp. 2071–2080, 2016.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Zhang, Y., Chen, X., Yang, Y., Ramamurthy, A., Li, B., Qi, Y., and Song, L. Can graph neural networks help logic reasoning?, 2019.

6. Appendix

A. Algorithm for Potential Useful Hidden Triples Identification.

According to their positions, "potential useful" triples can be divided into two categories: (1) triples that are the first or the last predicate in a ground Horn rule; and (2) triples that are neither the first nor the last. We proposed algorithms to identify both type of "potential useful" triples respectively, by taking the Horn rule in Eq. (2) as an example. Notation wise, we denote the $|E| \times |E|$ adjacent matrix associated with with each relation r_k in KG as $\mathbf{M}^k$, in which the element $\mathbf{M}_{ij}^k = 1$ if the triple $(e_i, r_k, e_j) \in O$, and zero otherwise.

- When the "potential useful" triple is the first or the last predicate in a ground Horn rule (i.e., the "potential useful" triple is $r_1(e_i, e_p)$ or $r_3(e_q, e_j)$), other observed triples still constitute a complete path, which can be extracted efficiently by sparse matrix multiplication. For example, to identify the "potential useful" triple $r_1(e_i, e_p)$, we have to first extract all connected path $r_2(e_p, e_q) \wedge r_3(e_q, e_j)$ by calculating $\mathbf{M} = \mathbf{M}^{(2)}\mathbf{M}^{(3)}$, where $\mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$ are adjacency matrices corresponding to relations r_2 and r_3. Each nonzero element $\mathbf{M}_{pj}$ indicates a connected path between e_p and e_j. We denote all indexes correspond to nonzero rows in $\mathbf{M}$ as $\delta = \{p|(\sum_j \mathbf{M}_{pj}) \neq 0\}$, which indicates that there is always a connected path starting at p. For specific $p \in \delta$, $\Delta_p = \{(e_i, r_1, e_p)|e_i \in E\}$ defines a set "potential useful" triples. If (e_i, r_1, e_p) in Δ_p is predicted to be true via KGE, the head predicates $r_0(e_i, e_j)$ can be inferred.
- Otherwise, the path corresponds to the conjunctive body of the ground Horn rule get broken into two paths by the "potential useful" triple, which we have to extract separately. For example, to identify "potential useful" triples $r_2(e_p, e_q) \in v_H$, two paths are essentially two single relations, whose corresponding matrices are $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(3)}$, respectively. We denote all indexes correspond to nonzero columns in $\mathbf{M}^{(1)}$ as $\delta_1 = \{p|(\sum_i \mathbf{M}_{ip}^{(1)}) \neq 0\}$ and all indexes correspond to nonzero rows in $\mathbf{M}^{(3)}$ as $\delta_2 = \{q|(\sum_j \mathbf{M}_{qj}^{(3)}) \neq 0\}$. $\Delta_{12} = \{(e_p, r_2, e_q)|p \in \delta_1, q \in \delta_2\}$ defines a set "potential useful" triples. If (e_p, r_2, e_q) in Δ_{12} is predicted to be true via KGE, the head predicates $\{r_0(e_i, e_j)|\mathbf{M}_{ip}^{(1)} \neq 0, \mathbf{M}_{qj}^{(3)} \neq 0\}$ can be inferred.

Note that a dynamic programming algorithm can be used to alleviate the computational complexity for long Horn rules.

B. Experimental Setting

Data Statistics We evaluate UniKER on both small experimental datasets and large scale real-world knowledge graph. To be specific, we include three small experimental datasets in total. They are RC1000, sub-YAGO3-10 and sub-Kinship. Since sub-Kinship is a subset of Kinship dataset, we will discuss it when we introduce Kinship dataset.

- RC1000 is a typical benchmark dataset for inference in MLN. It involves the task of relational classification.
- sub-YAGO3-10 is a subset of a well known benchmark dataset of knowledge graph, YAGO3-10.

For the large scale knowledge graph, we adopt three commonly used benchmark datasets, including Kinship, FB15k-237 and WN18RR.

- Kinship contains kinship relationships among members of a family (Denham, 1973). We substract a subset from Kinship dataset and call it sub-Kinship.
- FB15k-237 is the most commonly used benchmark knowledge graph datasets introduced in (Bordes et al., 2013). It is an online collection of structured data harvested from many sources, including individual, user-submitted wiki contributions.
- WN18RR is another widely used benchmark knowledge graph datasets introduced in (Bordes et al., 2013). It is designed to produce an intuitively usable dictionary and thesaurus, and support automatic text analysis. Its entities correspond to word senses, and relationships define lexical relations between them.

Compared Methods. We evaluate our proposed method against a number of state-of-the-art algorithms, including basic KG embedding models (e.g., RESCAL (Nickel et al., 2011), TransE (Bordes et al., 2013), DistMult (Toutanova et al., 2015)

Dataset	Type	#Entity	#Relation	#Triple	#Rule	Rule Generator
RC1000	Citation network	656	4	1006	3	hand-coded
sub-Kinship	Kinship network	68	12	412	41	hand-coded
sub-YAGO3-10	YAGO knowledge	55	8	61	5	AMIE+
Kinship	Kinship network	3007	12	28356	41	hand-coded
FB15k-237	Freebase knowledge	14541	237	310116	300	AMIE+
WN18RR	Lexical network	40943	11	93003	11	AMIE+

Table 3. Data Statistics.

and SimplE (Kazemi & Poole, 2018)), traditional logical rule-based algorithms (e.g., MLN (Richardson & Domingos, 2006) and BLP (De Raedt & Kersting, 2008)) and both classes of approaches to combine embedding model with logical rules. As for the approaches which design logical rules-based regularization to embedding models, we choose two representative methods to compare with, including KALE (Guo et al., 2016) and RUGE (Guo et al., 2017). For the approaches which design embedding-based variational distribution for variational inference of MLN, we compare with pLogicNet (Qu & Tang, 2019), ExpressGNN (Zhang et al., 2019) and pGAT (Harsha Vardhan et al., 2020).

Experimental Setup. To generate candidate rules, we hand-code logical rules for Kinship and RC1000 datasets, and mine rules on FB15k-237, WN18RR and sub-YAGO3-10 using AMIE+ (Galárraga et al., 2015). TransE (Bordes et al., 2013) and DistMult (Toutanova et al., 2015) are implemented as the score function for UniKER.