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

Kewei Cheng
Department of Computer Science, University of California, Los Angeles
viviancheng@cs.ucla.edu

Ziqing Yang
Department of Computer Science, Peking University
yangziqing@pku.edu.cn

Ming Zhang
Department of Computer Science, Peking University
mzhang@net.pku.edu.cnt

Yizhou Sun
Department of Computer Science, University of California, Los Angeles
yzsun@cs.ucla.edu

ABSTRACT

Knowledge graph inference has been studied extensively due to its wide applications in different domains. There are two main directions in solving the inference problem, i.e., logical rule reasoning and knowledge graph embedding (KGE). Logical rule-based approaches have shown their effectiveness due to the power of symbolic reasoning but suffer from low coverage, noise-sensitive, and scalability issues. KGE methods have demonstrated their good scalability when coping with large scale real-world KGs but fail to capture high-order dependency between entities and relations. Several attempts have been made to combine KG embedding and logical rules for better knowledge graph inference. Unfortunately, these approaches employ sampling strategies to randomly select only a small portion of ground rules or hidden triples, thus 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.

ACM Reference Format:
Kewei Cheng, Ziqing Yang, Ming Zhang, and Yizhou Sun. 2020. UniKER: A Unified Framework for Combining Embedding and Horn Rules for Knowledge Graph Inference. In *Proceedings of The Second International Workshop on Deep Learning on Graphs: Methods and Applications (DLG-KDD'20).* , 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Knowledge Graphs (KGs), such as Freebase [3], DBPedia [2], and YAGO [29], have grown rapidly in recent years, which provide extremely useful resources for many real-world applications, e.g., query expansion, information retrieval, and question answering [12, 18, 35, 38]. However, the extensive power of KGs is inevitably limited by the incompleteness of the real-world data. Consequently, KG reasoning, which aims at inferring missing knowledge through the existing facts, has gained increasing attention.

Knowledge graph embedding (KGE) currently hold the state-of-the-art in KG reasoning for their good scalability and strong robustness [4, 16, 27, 34, 36]. Despite the excellent performance of KGE methods, the ignorance of the symbolic compositionality of KG relations limits their application in more complex reasoning tasks. In addition, KGE methods highly rely on structure information in observed triple facts, which suffer from severe sparsity in real-world. Gathering more reliable triple facts becomes the key to improve their performance. An alternative solution is to infer missing facts via logical rules, which has been extensively explored by traditional logical rule-based methods [6, 23, 26]. Unlike KGE methods, logical rule-based methods are able to capture high-order dependency between entities and relations explicitly. However, a central challenge of such approach is the high computation complexity. Additionally, logical rule cannot handle noisy data due to its symbolic nature. Further, the performance of logical inference highly depends on the coverage of logical rules and the completeness of KGs, which suffer from severe insufficiency in reality.

Although both embedding-based methods and logical rule-based methods have their own limitations, they are complementary for better reasoning capability. On one hand, logical rules are useful to gather more reliable triples for KGE by exploiting symbolic compositionality of KG relations. On the other hand, high quality embedding learned by KGE models, in turn, will help to prepare cleaner and more complete KGs via excluding the noisy triples and inferring more facts for logical rule-based reasoning. Despite several attempts made to combine the advantage of embedding and logical rules for knowledge graph inference, most of them [8, 13, 24] only make a one-time injection of logic rules to KG embedding and thus fail to capture the mutual interaction between KG embedding and logical rules [13, 24]. 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.

DLG-KDD'20, August 24, 2020, Virtual Conference

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

To address the above issues, we propose a novel framework, UniKER, to combine embedding 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. The main contributions of this paper are summarized as follows:

- We investigate the problem of combining embedding and Horn rules for KG inference.
- A novel unified framework, UniKER, is proposed, which provides a simple yet effective iterative mechanism to let logical rule reasoning and knowledge graph embedding mutually enhance each other in an extremely efficient way.
- Our UniKER also shows robustness to noise in KGs, which previous methods cannot cope with.
- We experimentally demonstrate that UniKER is superior to existing state-of-the-art algorithms in terms of both efficiency and effectiveness.

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. Every relation r_k is associated with an $|E| \times |E|$ matrix $\mathbf{M}^k$, in which the element $\mathbf{M}^k_{ij} = 1$ if the triple $(e_i, r_k, e_j) \in O$, and zero otherwise. 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 Friend$(\cdot, \cdot)$. By assigning constants Alice and Bob to it, we get a ground predicate Friend(Alice, Bob). A 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. FOL rules 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 [7] and AMIE [10, 11] 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:

$$r_0(x, y) \leftarrow r_1(x, z_1) \wedge r_2(z_1, z_2) \wedge r_3(z_2, y) \tag{1}$$

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:

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

A Brief Review over Knowledge Graph Inference There are two main directions in solving the KG inference problem, i.e., traditional logical inference and knowledge graph embedding (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 [1]. One approach to this problem is stochastic local search, exemplified by WalkSAT [26]. Markov Logic Network (MLN) [23] 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 [4], TransH [33], TransR [17], DistMult [36], ComplEx [32], SimplE [16] and RotatE [30], which differ from each other with different SFs.

Existing work on Integrating Logical Rules and KGE Existing methods of integrating logical rules and KGE can be broadly divided into two categories: (1) designing logical rule-based regularization to embedding models and (2) designing embedding-based variational distribution for variational inference of MLN.

Logical Rule-based Regularization in Embedding Loss. The first way to combine two worlds is to 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 [13], RUGE [14] and Rocktäschel et al. [24] are the representative methods. One main drawback is that they have to instantiate universally quantified rules into ground rules before learning models, which is extremely time-consuming. Therefore, a sampling strategy is usually employed by selecting a small portion of ground rules to approximate the inference process, which inevitably leads to the loss of information from the logical rule side. In addition, most methods in this category make only one-time injection of logical rules to enhance embedding, ignoring the interactive nature between embedding learning and logical inference [13, 24].

Embedding-based Variational Inference for MLN.. Several methods including pGAT [15], ExpressGNN [39] and pLogicNet [22] propose to leverage graph embedding to define variational distribution for all possible hidden triples to conduct variational inference of MLN. Although they provide an elegant approach to integrate embedding models and logical rules in a single representation by optimizing MLN using variational inference methods, inference efficiency is still a central challenge. Given the fact that KGs usually contain over ten thousand entities, it is impractical to optimize

over all the hidden triples. Therefore, all approaches in this categories only sample a small portion of hidden triples to reduce the computational complexity. This brings in the similar issue of information loss from the logical rule side, as selecting only a small portion of hidden triples will inevitably results in the ignorance of a significant number of ground rules.

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.

3.1 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 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. □

3.2 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. The next question is how to conduct such assignment efficiently. We denote the satisfying truth assignment 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** [25] 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 predicates into calculation, forward chaining adopts "lazy inference" instead. It calculates only a small subset of "active" groundings of predicates/rules, and activates more if necessary as the inference proceeds. The mechanism dramatically improves inference efficiency via avoiding the computation for a large number of ground predicates/rules that are never used. Moreover, considering that the conjunctive body of a ground Horn rule is essentially a path in KGs, which can be extracted efficiently using sparse matrix multiplication, forward chaining can also be implemented efficiently.

3.3 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 incompleteness of the KG and the errors contained in KG. Considering 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. Since the body of a Horn rule is a conjunction of predicates, a ground Horn rule can get activated and contribute to logical inference only if all the predicates in its body are completely observed. Due to the sparsity of real-world KGs, only a small portion of ground Horn rules can participate in logical inference, 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_+. 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.

- 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 $\delta_1 = \{p | (\sum_i \mathbf{M}^{(1)}_{ip}) \neq 0\}$ and all indexes correspond to nonzero rows in $\mathbf{M}^{(3)}$ as $\delta_2 = \{q | (\sum_j \mathbf{M}^{(3)}_{qj}) \neq 0\}$. $\Delta_{12} = \{(e_p, r_2, e_q) | p \in \delta_1, q \in \delta_2\}$ 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 $\{r_0(e_i, e_j) | \mathbf{M}^{(1)}_{ip} \neq 0, \mathbf{M}^{(3)}_{qj} \neq 0\}$ can be inferred.

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, a cleaner KG is significant for logical inference. Additionally, although we assume that logical rules are entirely correct under any circumstance, it is not true in the real world. Logical rules themselves could also be uncertain and thus bring potential errors to inference results. To relieve the uncertainty brought by logical rules, it is necessary to check the correctness of the inferred triples before add them to the KGs. 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.

3.4 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:

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

where a common margin-based pairwise ranking loss is employed to define the objective function. When learning the entity and relation embeddings, we treat triple (e_i, r_k, e_j) in both O and $\mathbf{v}_H^{T^*}$ as positive example while (e_i', r_k, e_j') is its corresponding negative sample, 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 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.

Algorithm 1 Learning Procedure for UniKER

Require: Observed facts in knowledge bases O
Require: Threshold used to eliminate noise $\theta\%$
Require: A set of Horn rules $\mathcal{F}$
1: **while** more triples can be derived **do**
2: Derive $\mathbf{v}_H^{T\,i*}$ from O then update $O \leftarrow O \cup \mathbf{v}_H^{T\,i*}$
3: Train KGE based on O.
4: Compute Δ_- then update $O \leftarrow O - \Delta_-$.
5: Compute Δ_+ according to Section 3.3 then update $O \leftarrow O \cup \Delta_+$.
6: **end while**

3.5 Integrating Embedding and Logical Rules in an Iterative Manner.

Since logical rules and KG embedding can mutually enhance each other as discussed above, we propose a unified framework, known as UniKER, to integrate KG embedding 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 a subset of $\mathbf{v}_H^{T^*}$ as $\mathbf{v}_H^{T\,i*}$ 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 and add new triples Δ_+, which are potentially useful according to Section 3.3. The pseudo-code of the iterative learning procedure of UniKER can be found in Algorithm 1.

4 EXPERIMENTS

4.1 Experimental Setting

Datasets. We implement our experiments on three small scale datasets (i.e., RC1000, sub-YAGO3-10 and sub-Kinship) and three large scale real-world KGs (i.e., Kinship, FB15k-237 and WN18RR). The detailed statistics of the datasets are provided in appendix.

Compared Methods. We evaluate our proposed method against state-of-the-art algorithms, including basic KGE models (e.g., RESCAL [19], TransE [4], DistMult [31] and SimplE [16]), traditional logical rule-based algorithms (e.g., MLN [23] and BLP [6]) and both classes of approaches to combine embedding model with logical rules. For approaches that design logical rule-based regularization to embedding

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

Model	Kinship			FB15k-237			WN18RR		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
RESCAL	0.489	0.894	0.639	0.108	0.322	0.179	0.123	0.239	0.162
SimplE	0.335	0.888	0.528	0.150	0.443	0.249	0.290	0.351	0.311
KALE	0.433	0.869	0.598	0.131	0.424	0.230	0.032	0.353	0.172
RUGE	0.495	0.962	0.677	0.098	0.376	0.191	0.251	0.327	0.280
BLP†	-	-	-	0.062	0.150	0.092	0.187	0.358	0.254
MLN†	0.655	0.732	0.694	0.067	0.160	0.098	0.191	0.361	0.259
ExpressGNN	0.105	0.282	0.164	0.150	0.317	0.207	0.036	0.093	0.054
pLogicNet†	0.683	0.874	0.768	0.237	0.524	0.332	0.398	0.537	0.441
pGAT‡	-	-	-	0.377	0.609	0.457	0.395	**0.578**	0.459
TransE†	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†	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 [22].
‡ Results are taken from [15].

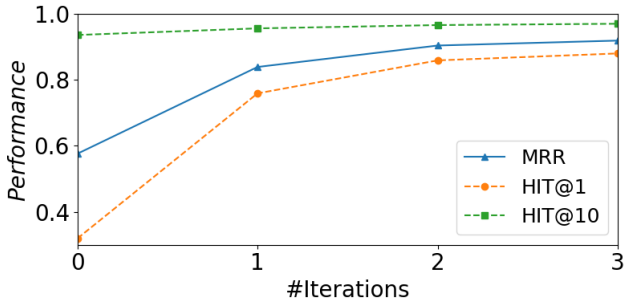

Figure 1: Performance of KG Completion on Kinship Dataset w.r.t. #Iterations for Effectiveness Analysis.

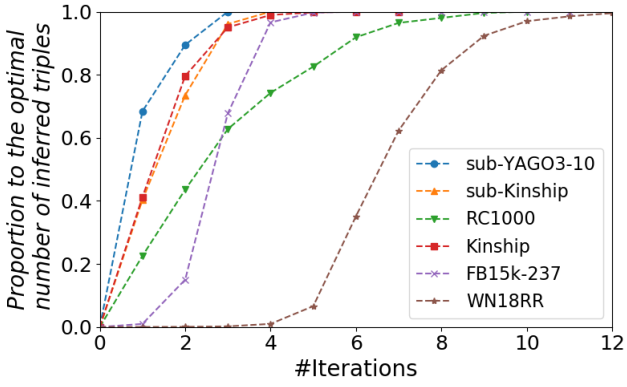

Figure 2: Proportion to the Optimal Number of Inferred Triples w.r.t. #Iterations for Efficiency Analysis.

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. θ% is the Threshold Used to Eliminate Noise.

models, we choose two representative methods to compare with, i.e., KALE [13] and RUGE [14]. For those which design embedding-based variational distribution for variational inference of MLN, we compare with pLogicNet [22], ExpressGNN [39] and pGAT [15].

Experimental Setup. To generate candidate rules, we hand-code logical rules for Kinship and RC1000 datasets, and automatically mine rules on FB15k-237, WN18RR and sub-YAGO3-10 using AMIE+ [10]. TransE [4] and DistMult [31] are implemented as the scoring function for UniKER.

4.2 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 [4] and three evaluation metrics, i.e., Hit@1, Hit@10 and Mean Reciprocal Rank (MRR). More detailed settings are in appendix. 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 and added to the observed facts. To investigate how this iterative process helps improve reasoning ability of UniKER, we conduct experiments on Kinship dataset and record the performance of UniKER on test data in terms of Hit@1, Hit@10 and MRR in every iteration. In particular, iteration 0 represents the step at the very beginning, where 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, the improvement slows down gradually; (2) UniKER has a bigger impact on Hit@1, Hit@10 compared to MRR.

Robustness Analysis. In addition to the strong reasoning ability, our UniKER also shows robustness for noisy KGs. To investigate the robustness of UniKER, we compare the reasoning ability of UniKER with TransE on Kinship dataset with noise. As all kinds of noise might be contained in the process of constructing KGs, 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 data. All generated noisy triples only fused into the original training set while validation and test sets remain the same. Moreover, 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.

4.3 Efficiency Analysis

Since forward chaining learns the optimal truth assignment for HORN-SAT problem iteratively, the number of iterations required to achieve the optimal solution may influence its scalability. We conduct two experiments on all the six datasets: (1) as presented in Figure 2, we record the proportion of inferred triples accumulated in every iteration over all inferred triples. The result shows that forward chaining could achieve the optimal solution within 12 iterations, and achieve majority of correct triples only within 4 iterations; (2) as illustrated in appendix, we evaluate the scalability of forward chaining against a number of state-of-the-art inference algorithms for MLN (e.g., MCMC [5], MC-SAT [21], BP [37], lifted-BP [28] and Tuffy [20]). Forward chaining runs $100 - 100,000$ times faster than them. Some widely used algorithms MCMC and MC-SAT even cannot handle RC1000 dataset, which indicates the scalability of UniKER.

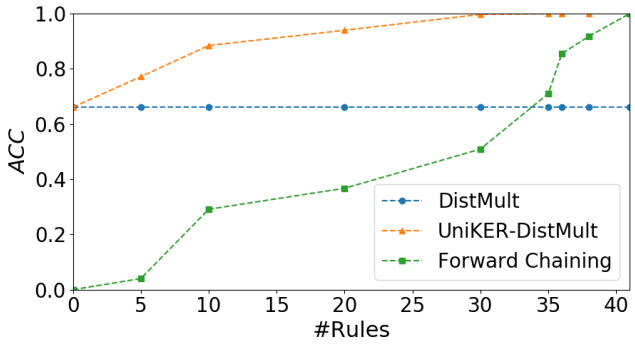

Figure 3: Impact of Coverage of Logical Rules on Kinship Dataset on Triple True/False Classification Task.

4.4 Impact of Coverage of Logical Rules on Kinship Dataset

We also conduct experiments to analyze the impact of coverage of logical rules on KG inference. Due to space limitation, we only show the results on Kinship dataset as we have similar observations on the remaining datasets. We compare UniKER-DistMult against its default model DistMult as well as forward chaining algorithm over different coverage, which is measured by the total number of triples inferred from the given set of Horn rules. And we evaluate all methods in terms of triple True/False classification accuracy, regarding link prediction in KG as binary classification. The number of triples inferred from different sets of rules are provided in appendix. As shown in Figure 3, we find that: (1) When the coverage is not enough, traditional rule-based methods show poor performance; (2) Without incorporating with logical rules, DistMult has already shown pretty good reasoning ability; (3) The performance of UniKER steadily increases with the increase of the coverage of logical rules; (4) UniKER has already achieved accuracy close to 1 with only 30 rules, which is much higher than forward chaining. The small number of logical rules is appealing in practice as it is costly and labor-intensive to obtain high-quality logical rules.

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

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

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

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

[5] C. M. Carlo. Markov chain monte carlo and gibbs sampling. *Lecture notes for EEB*, 581, 2004.

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

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

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

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

[10] L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek. 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.

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

[12] J. Graupmann, R. Schenkel, and G. Weikum. The SphereSearch engine for unified ranked retrieval of heterogeneous XML and web documents. In *Proceedings of the 31st international conference on very large data bases*, pages 529–540. VLDB Endowment, 2005.

[13] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo. Jointly embedding knowledge graphs and logical rules. pages 192–202, 01 2016.

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

[15] L. V. Harsha Vardhan, G. Jia, and S. Kok. Probabilistic logic graph attention networks for reasoning. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 669–673, New York, NY, USA, 2020. Association for Computing Machinery.

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

[17] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[18] D. Lukovnikov, A. Fischer, J. Lehmann, and S. Auer. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, pages 1211–1220. International World Wide Web Conferences Steering Committee, 2017.

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

[20] F. Niu, C. Ré, A. Doan, and J. Shavlik. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *Proceedings of the VLDB Endowment*, 4(6):373–384, 2011.

[21] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *AAAI*, volume 6, pages 458–463, 2006.

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

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

[24] T. Rocktäschel, S. Singh, and S. Riedel. 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*, pages 1119–1129, 2015.

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

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

[27] Y. Shan, C. Bu, X. Liu, S. Ji, and L. Li. Confidence-aware negative sampling method for noisy knowledge graph embedding. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 33–40. IEEE, 2018.

[28] P. Singla and P. M. Domingos. Lifted first-order belief propagation. In *AAAI*, volume 8, pages 1094–1099, 2008.

[29] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

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

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

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

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

[34] R. Xie, Z. Liu, F. Lin, and L. Lin. Does william shakespeare really write hamlet? knowledge representation learning with confidence. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[35] C. Xiong, R. Power, and J. Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pages 1271–1279. International World Wide Web Conferences Steering Committee, 2017.

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

[37] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in neural information processing systems*, pages 689–695, 2001.

[38] W. Yih, M.-W. Chang, X. He, and J. Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *IJCNLP*, pages 1321–1331, Beijing, China, July 2015.

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

6 APPENDIX

A DATA STATISTICS

Table 3: Data Statistics.

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+

The detailed statistics of three small scale datasets (e.g., RC1000, sub-YAGO3-10 and sub-Kinship) and three large scale real-world KGs (e.g., Kinship, FB15k-237 and WN18RR) are provided in Table 3. 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, i.e., 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 [9]. 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 [4]. 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 [4]. 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.

B EXPERIMENTAL DETAILS.

B.1 Setting for Knowledge Graph Completion

To compare among the reasoning ability of UniKER and aforementioned baseline algorithms, we mask the head or tail entity of each test triple, and require each method to predict the masked entity. We use three large-scale datasets including Kinship, FB15K-237 and WN18RR. During evaluation, we use the filtered setting [4] and three evaluation metrics, i.e., Hit@1, Hit@10 and MRR. We take the results of BLP, MLN, TransE, DistMult and pLogicNet from the corresponding paper [22] and take the results of pGAT from the corresponding paper [15]. As only the results on the FB15k-237 and WN18RR datasets are reported, we only compare with them on these two datasets. The results of MLN, TransE, and DistMult on Kinship dataset are reported by us.

B.2 Efficiency Analysis

Table 4: Comparison of Inference Time for Forward Chaining vs. MLN.

Model	sub-YAGO3-10	sub-Kinship	RC1000	Kinship	FB15k-237	WN18RR
MCMC	76433s	-	-	-	-	-
MCSAT	1292s	25912s	-	-	-	-
BP	10s	16343s	-	-	-	-
liftedBP	15s	16075s	-	-	-	-
Tuffy	0.849s	1.398s	4.899s	-	-	-
Forward Chaining	0.003s	0.034s	0.007s	0.593s	186s	30s

We evaluate the scalability of forward chaining against a number of state-of-the-art inference algorithms for MLN (e.g., MCMC [5], MC-SAT [21], BP [37], liftedBP [28] and Tuffy [20]) over all the six datasets given in Table 3. Details of each baseline are listed below:

- Markov Chain Monte Carlo (MCMC) [5] is the most widely used method for approximate inference in MLN. The basic idea is to utilize Markov blanket to calculate marginal probabilities of ground predicates.

- MC-SAT [21] is an efficient MCMC algorithm that combines slice sampling with satisfiability testing.
- Belief Propagation (BP) [37] is another widely used method for approximate inference in MLN. It is a message-passing algorithm for performing inference on graphical models.
- lifted Belief Propagation (liftedBP) [28] is a lifted version of belief propagation algorithm.
- Tuffy [20] is an open-source MLN inference engine that achieves scalability compared to prior art implementations.

The results of their inference time are given in Table 4.

B.3 Impact of Coverage of Logical Rules on Kinship Dataset

Table 5: Coverage of Different Number of Rules on Kinship Dataset.

#Rule	5	10	20	30	35	36	38	41
#Inferred Triple	870	6276	7915	10973	15302	18429	19780	21549

To ensure enough coverage of logical rules, we take the whole Kinship dataset as training data while the triples, which can be inferred from the training data using all 41 logical rules, are regarded as test data. We vary the number of Horn rules among $\{10, 20, 30, 35, 36, 38, 41\}$ to investigate the effects of coverage of logical rules and the number of triples that can be inferred from these sets of rules is shown in Table 5.