

Article

HRER: A New Bottom-Up Rule Learning for Knowledge Graph Completion

Zongwei Liang ^{*,†} , Junan Yang [†], Hui Liu, Keju Huang, Lin Cui, Lingzhi Qu and Xiang Li

College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China; yangjunan@ustc.edu (J.Y.); christ592604@163.com (H.L.); huangkeju@163.com (K.H.); cuilin17@nudt.edu.cn (L.C.); qulingzhi@nudt.edu.cn (L.Q.); lix20@nudt.edu.cn (X.L.)

* Correspondence: zwliang17@nudt.edu.cn

† These authors contributed equally to this work.

Abstract: Knowledge graphs (KGs) are collections of structured facts, which have recently attracted growing attention. Although there are billions of triples in KGs, they are still incomplete. These incomplete knowledge bases will bring limitations to practical applications. Predicting new facts from the given knowledge graphs is an increasingly important area. We investigate the models based on logic rules in this paper. This paper proposes HRER, a new bottom-up rule learning for knowledge graph completion. First of all, inspired by the observation that the known information of KGs is incomplete and unbalanced, HRER modifies the indicators for screening based on the existing relation rule mining methods. The new metric *HRR* is more effective than traditional confidences in filtering Horn rules. Besides, motivated by the differences between the embedding-based methods and the methods based on logic rules, HRER proposes entity rules. The entity rules make up for the limited expression of Horn rules to some extent. HRER needs a few parameters to control the number of rules and can provide the explanation for prediction. Experiments show that HRER achieves the state-of-the-art across the standard link prediction datasets.

Keywords: NLP; knowledge graphs; link prediction; reasoning; uneven distribution; reliability; horn rule; entity rule



Citation: Liang, Z.; Yang, J.; Liu, H.; Huang, K.; Cui, L.; Qu, L.; Li, X. HRER: A New Bottom-Up Rule Learning for Knowledge Graph Completion. *Electronics* **2022**, *11*, 908. <https://doi.org/10.3390/electronics11060908>

Academic Editor: Wojciech Mazurczyk

Received: 16 February 2022

Accepted: 11 March 2022

Published: 15 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Large scale knowledge graphs (KGs) such as Freebase [1], DBpedia [2], NELL [3], and YAGO [4], have achieved significant development in recent years. These KGs contain considerable facts stored in the form of triples (h, r, t) , where h, r, t represents the heads, the relations, and the tails, respectively. KGs play a crucial role in intelligent question answering, search engines, and smart healthcare applications. However, KGs cannot exhaust all triples. Although there are billions of triples in KGs, they are still incomplete. The incomplete KGs will cause limitations to practical applications. For example, over 70% of people included in Freebase have no known place of birth, and 99% have no known ethnicity, which will significantly limit our search and answering. Therefore, knowledge graph reasoning, which infers new knowledge based on incomplete KGs, has received increasing attention.

Link prediction is a fundamental task of knowledge graph reasoning. Link prediction means that, given $(?, r, t)$, predict the missing head entity based on the existing knowledge graphs or given $(h, r, ?)$, predict the missing tail entity based on the existing knowledge graphs, where $?$ represents unknown entities. This paper divides the methods of link prediction into two types based on the characterization [5–7]: one is the embedding methods, which utilize latent features, and the other is the traditional methods based on logic rules which employ observed features.

Embedding methods are mainstream approaches [1,8–15]. These methods learn the embeddings of entities and relations simultaneously, then measure the rationality of triples

through specific score functions between entities and relations. These methods can achieve better performances because they are free from the restrictions of rule representation. However, these methods have two shortcomings. The first weakness is the unexplainable inference, which means that the error prediction cannot be modified in practical application. Another defect is that these models have different parameters and are sensitive to specific parameters, making them hard to compare the pros and cons.

Logic-rule methods originated from inductive logic programming (ILP) in semiotics. Although these methods [5,16–18] do not perform well on standard datasets, practical applications are more inclined to adopt these methods for their interpretability. In practical applications [5–7], people can artificially modify the biased results of the interpretable models.

In this paper, our overarching interest is explainable inferences, i.e., the method of logic rules. However, the current logic-rule methods have the following two drawbacks.

The first one is that the models ignore the incomplete and unbalanced facts in KGs. Most traditional studies were designed based on the closed world assumption, such as Standard Confidence. However, the KGs are the open domain datasets, which may have more incomplete facts in open domain knowledge bases. Although some methods, such as AMIE [5] and RuleN [18], consider this incompleteness, their rule metrics are still inappropriate. These inappropriate metrics severely limit the number and quality of logic rules mined, which drives us to design more reasonable metrics.

Besides, the second one is that the limited characterizable rules restrict the performance. The current methods only mine the Horn rules, which are relatively simple and cannot describe the complicated relations of entities. The limited representation limits the model, thus we need to mine other non-Horn rules

In order to alleviate the above two drawbacks of the current logic-rule methods, this paper proposes HRER, a knowledge graph reasoning model based on the logic rule and entity rules. The specific strategies of HRER to these drawbacks are as follows.

First of all, since current filtering indicators ignore the incompleteness and biased distribution of information in KGs, we propose a new index—Horn rule reliability (*HRR*). The index *HRR* takes into account the incompleteness of the knowledge graph and the biased distribution of the facts, which can screen Horn rules more reasonably. Experiments on benchmark datasets show that the indicator *HRR* performs better in mining Horn rules.

Besides, to solve the problem that the limited form of logic rules restricts the performance, this paper first analyzes the differences between the logic rule methods and the embedding-based methods. Embedding-based methods learn to represent entities and relations simultaneously, which learn the equivalence of relations and the relevance of entities. Inspired by this observation, this paper proposes entity rules based on Horn rules. Entity rules can mine the inclusion and equivalence relations of entities on attributes. Experiments on benchmark datasets show that entity rules can effectively improve the performance of models based on Horn rules.

As shown in Figure 1, this is the architecture of HRER. HRER contains two main parts. The first part is relation rules (i.e., Horn rules) mining, and the second part involves mining entity rules. The upper part of Figure 1 shows the mining of Horn rules with Horn rule reliability and the lower part of Figure 1 shows the searching of entity rules. Then we achieve special weights of two types of rules based on the overall performance of training data. Finally, we apply the special weights to merge the two types of rules and perform link predictions.

In summary, the main contributions of this paper are as follows.

- This paper proposes the new index—Horn rule reliability (*HRR*), which alleviates the problem caused by incompleteness and biased distribution. Experiments show that the Horn rule based on this metric *HRR* achieves state-of-the-art performance in the link prediction task.
- This paper proposes the reasoning of entity rules, which makes up for insufficient representation of relation rules to some extent. Experiments show that the inference based on entity rules can improve link prediction by at least 2% on Hit@10.

- HRER is explainable, providing the basis for the prediction. Unlike the embedding models, which are sensitive to parameters, HRER has only a few parameters for controlling the number of rules.

The rest of this paper is structured as follows. Section 2 presents a brief overview of related work. Section 3 introduces the problem formulation, including definitions and preliminaries. Section 4 is the central part of the model, which mainly explains the design of the Horn rule reliability index and the realization of entity rules. Section 5 describes the new evaluation metric and experiments. Finally, we summarize our findings along with the future directions in Section 6.

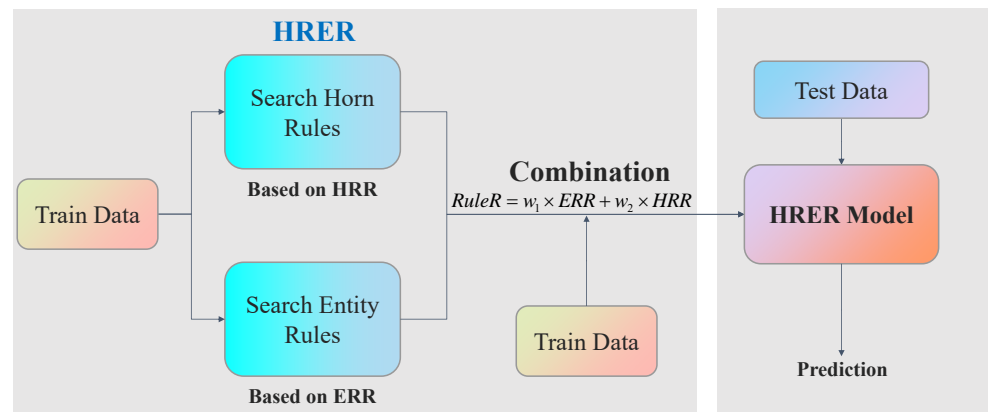


Figure 1. Model Architecture.

2. Related Work

This section describes related works and the critical differences between them. We divide knowledge graph models into two families: methods based on latent features and methods based on observed features.

2.1. Methods Based on Latent Features

Methods based on latent features (i.e., the embedding methods) belong to numerical reasoning. These methods first design the corresponding representation (including the representation of entities and relations, and the score function of triples) and train to make the matching score of the correct triples get the maximum value (i.e., make the matching function of the error triples achieve the minimum). Finally, the trained models are applied for link prediction. We divide embedding methods into three types according to the matching function: geometric models, tensor factorization models and deep learning models.

Geometric Models utilize the relations as the transform between heads and tails in latent spaces. TransE [1] directly selects the euclidean distance of the entities and the relation vectors to measure the matching degree of triples. However, TransE cannot describe the 1-N, N-1, N-N relationship well. TransH [19] revises the representation of entities and proposed that entities have different representations in different relations. TransR [12] thinks that using the same semantic space cannot adequately represent knowledge. TransR imports a mapping matrix to map entity vectors to different attribute spaces. RotatE [14] proposes the rotation of complex vectors to characterize the rules between relations better. Inspired by the fact that concentric circles in the polar coordinate system can naturally reflect the hierarchy, HAKE [20] maps entities into the polar coordinate system. HAKE can effectively model the semantic hierarchies in knowledge graphs.

Tensor Factorization Models define the dot product of tensors as the matching function. RESCAL [21] designs the relations as the full rank matrixes. On this basis, DistMult [22] proposes the mapping matrix to be a diagonal matrix. ANALOGY [23] improves the mapping matrix to a standard matrix. ComplEx [15] introduces complex-valued matrixes to represent the relations based on DistMult, which describes the asymmetric and anti-

sense rules better. TuckER [13] imports tucker decomposition, and this model achieves state-of-the-art results on some datasets. Simple [24] is a simple enhancement of CP to allow the two embeddings of each entity to be learned dependently. HoIE [25] is a multiplicative model that is isomorphic to ComplEx [15]. Inspired by the recent success of automated machine learning (AutoML), AutoSF [26] proposes to automatically design scoring functions for distinct KGs by the AutoML techniques.

Deep Learning Models use deep neural networks to perform knowledge graph completion. ConvE [27] and ConvKB [28] employ convolutional neural networks to define score functions. CapsE [29] embeds entities and relations into one-dimensional vectors under the basic assumption that different embeddings encode homologous aspects in the same positions. CompGCN [30] utilizes graph convolutional networks to update the knowledge graph embedding. Neural Tensor Network (NTN) combines E-MLP with several bilinear parts. Nathani [31] proposes a novel attention-based feature embedding that captures both entity and relation features in any given entity's neighborhood.

The representation of triples ranges from one-dimensional vectors to multi-dimensional tensors, and the matching function ranges from simple distance to hyperplane mapping. Together, these studies make improvements to have a better description of knowledge. Although such methods perform better on basic datasets, they are unexplainable and sensitive to parameters.

2.2. Methods Based on Observed Features

The methods based on observed features belong to symbolic reasoning. They mine relevant relation rules based on observable statistical features and then accomplish reasoning with these relation rules. Collectively, these methods generally apply association algorithms to mine Horn rules in the knowledge base, and there are also methods mining rules with experts.

Such methods originated from inductive logic programming. Sherlock [17] is a typically unsupervised method for mining logic rules. It extracts first-order Horn rules from network text and reasons with probabilistic graphical models (PGMs). Similar ILP methods are WARMR [16] and ALEPH [32]. However, these methods, which are not designed for open-domain knowledge bases, are not suitable for knowledge graph reasoning. Pang-Ning Tan [33] studies the association mining method, and most subsequent rules mining adopts this association method. PRA mines paths with a high probability of occurrence through random routing and incorporated path features as matrix features into machine learning. AMIE [5] is a typical method for mining association rules based on the partially complete assumption (PCA). RuleN [18] improves AMIE, which mines rules on the part of the knowledge base. RuleN integrates the method of path search and AMIE. AnyBURL [34,35] proposes an anytime bottom-up technique for learning logical rules from large knowledge graphs. AnyBURL [34,35] applies the learned rules to predict candidates in the context of knowledge graph completion.

Overall, these approaches mine the closed Horn rules existing in knowledge bases and use them to accomplish reasoning. These approaches perform poorly on standard datasets, but they are explainable.

3. Background

In this section, we introduce Horn rules and related concepts.

Let \mathcal{E} denote the set of all entities and \mathcal{R} the set of all relations present in KGs. In the following, we utilize the notation (h, r, t) (head entity, relation, tail entity) to identify a triple in KG, with $h, t \in \mathcal{E}$, $r \in \mathcal{R}$ denoting the subject(head), the object(tail) and the relation between them, respectively.

Horn Rule. An atom is a fact that has variables at the subject or object position. A Horn rule consists of a head and a body, where the head is a single atom and the body is a set of atoms. The paper denotes rule with head $r(x, y)$ and body $\{B_1, \dots, B_n\}$:

$$B_1 \wedge B_2 \wedge \dots \wedge B_n \Rightarrow r(x, y) \quad (1)$$

where B_1 represents the atom $r_1(x, z_1)$, B_i represents the atom $r_i(z_{i-1}, z_i)$ and B_n represents the atom $r_n(z_m, y)$. Horn Rules can be abbreviated as $\vec{B} \Rightarrow r(x, y)$. An instance of the rule is:

$$hasChild(p, c) \wedge isCitizen(p, s) \Rightarrow isCitizen(c, s) \quad (2)$$

In this paper, relation rules mined by HERE are Horn rules. We reason the head of the rule based on the body.

Support. The support of a rule quantifies the number of correct predictions, i.e., the number of distinct pairs of subjects and objects in the head. We calculate support as:

$$supp(\vec{B} \Rightarrow r(x, y)) := \exists z_1, \dots, z_m : \vec{B} \wedge r(x, y) \quad (3)$$

Head Coverage. Support is an absolute quantitative indicator. The same number of supports in knowledge bases with different scales have different meanings, so the literature [5] designs the relative indicator Head Coverage. Head Coverage is the proportion of pairs from the head relation that are covered by the predictions of the rule:

$$hc(\vec{B} \Rightarrow r(x, y)) := \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x', y') : r(x', y')} \quad (4)$$

with $\#(x', y') : r(x', y')$ as an abbreviation for $|\{(x', y') : x, y \in \mathcal{E}, r(x', y')\}|$.

Standard Confidence. The standard confidence measure takes all facts that are not in the KGs as negative evidence. Thus, the standard confidence of a rule is the ratio of its predictions that are in the KGs, i.e., the share of A in the set of predictions:

$$conf(\vec{B} \Rightarrow r(x, y)) := \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y) : \exists z_1, \dots, z_m : \vec{B}} \quad (5)$$

The standard confidence is blind to the distinction between “false” and “unknown”. Thus, it implements a closed world setting. It mainly describes the known data and penalizes rules that make a large number of predictions in the unknown region. Reasoning, in contrast, aims to maximize the number of true predictions that go beyond the current knowledge. We do not want to describe data but to predict data.

Partial Completeness. AMIE [5] proposes to generate negative evidence by the partial completeness assumption (PCA). This is the assumption that if $r(x, y)$ in $KBtrue$ for some x, y , then

$$\forall y' : r(x, y') \in KBtrue \cup NEWtrue \Rightarrow r(x, y') \in KBtrue \quad (6)$$

In other words, AMIE assumes that if the database knows some r -attribute of x , then it knows all r -attributes of x . This assumption is valid for functional relations r , such as birth dates, capitals, etc. These usually contain either all r -values or none for a given entity. The assumption is also valid in the vast majority of cases for relations that are not functional, but that have high functionality. Even for other relations, the PCA is still reasonable for knowledge bases that have been extracted from a single source (such as DBpedia and YAGO). These usually contain either all r -values or none for a given entity.

PCA Confidence. AMIE [5] proposes the partial completeness assumption: if specific R attributes about entity x appear in the given knowledge base, then the model assumes that the knowledge base includes all R attributes of x . There is no R attribute of entity x in the triples needed to be inferred. AMIE changes the denominator as the set of facts we know correct, together with the facts which we assume are false.

Under the PCA, AMIE [5] normalizes the confidence not by the entire set of facts but by the set of facts which we know are true, together with the facts which we assume are false. If the head atom of the rule is $r(x, y)$, then this set is just the set of facts $\{r(x, y') : r(x, y') \in \mathcal{K}\}$. Thanks to the FUN-Property, the PCA is always applied to the first argument of the head atom:

$$pcaconf(\vec{B} \Rightarrow r(x, y)) := \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y) : \exists z_1, \dots, z_m, y' : B \wedge r(x, y')} \quad (7)$$

AMIE. AMIE [36] implements rule mining through a parallel search, which has high computing efficiency. AMIE [36] utilizes language bias to limit the search space, i.e., each atom in the rule is related to other atoms through the head entity or tail entity. AMIE defines that a rule is closed if every variable appears at least twice. AMIE only mines closed rules. This paper mines Horn rules through the rule mining algorithm in AMIE.

4. HRER Model

This section introduces HRER—a knowledge graph reasoning model based on Horn Rule and Entity Rule. Section 4.1 introduces the overall framework of the model. Section 4.2 introduces the implementation method of relation rules, which mainly explains the key indicator: Horn rule reliability. Section 4.3 introduces the implementation of entity rules.

4.1. Model Overview

Figure 1 shows the overall implementation process of HRER. HRER consists of mining the Horn rules and the entity rules in KGs. Then we combine the two rules with different weights and perform the link prediction.

For the fast searching, we adopt the rule mining algorithm in AMIE [36] to mine Horn rules upon the Horn rule reliability (*HRR*), which will be described in detail in Section 4.2, in the first part of the rule mining step. The second part is similar to the first part. We apply association algorithms in mining entity rules. Finally, we combine two types of rules by special weights based on the overall performance of the training dataset and perform link prediction on the test dataset.

4.2. Reasoning Based on Relation Rules

As the common methods based on logic rules [36], the first step is mining the Horn rules. This paper only mines closed Horn rules with the associated algorithms. For a better understanding, we take the following 2-hop closed Horn rule as the instantiation analysis:

$$MotherOf(p, c) \wedge MarryTo(p, s) \Rightarrow FatherOf(c, s) \quad (8)$$

Figure 2 shows the mechanism of the association search method. By traversing all relations, we filter triples crossing heads/tails to build a closed-loop, i.e., the closed Horn rule needed to mine. The specific implementation can be seen in AMIE [36].

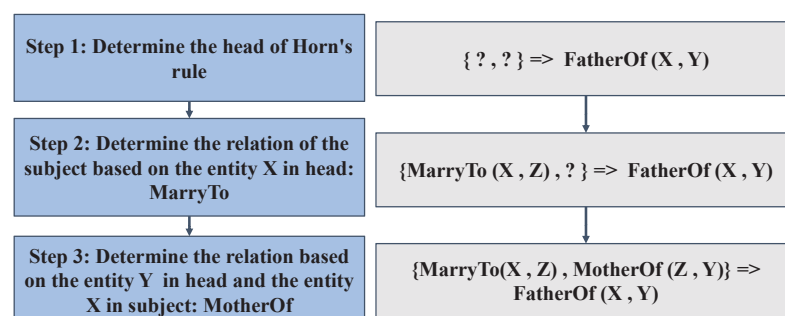


Figure 2. This is a toy example of mining closed Horn rules.

The reliability of Horn rules mined by different facts is different. For the excavated Horn rules, we need to measure the reliability of the rules. Traditional methods utilize indicators such as Standard Confidence and PCA Confidence to rank the rules. However, the design of these indicators do not consider the uncertainty of incomplete information and the uneven distribution of facts in KGs. The following part focuses on the motivation and the design of the index—Horn Rule Reliability (*HRR*).

The incompleteness of the KGs is uncertain. The existing KG commonly satisfies that the number of known triples is more than that of unknown triples. However, for some special logic rules, the number of unknown triples may exceed that of known triples. As shown in Figure 3, the area of unknown triples exceeds known triples in the KGs. The traditional rule indexes do not consider this phenomenon, and they utilize the total number of triples in KGs. For example, Standard Confidence considers all entity pairs involved in the relations, and the rationality of rules will reduce as the proportion of unknown knowledge increases.

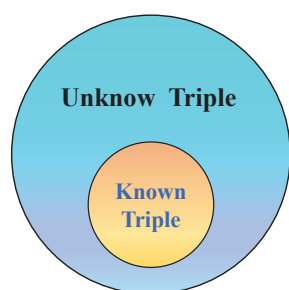


Figure 3. There may be more unknown triples in the knowledge base.

Given the uncertainty of the incomplete information in the knowledge base, this paper does not consider all the information of the head relation (i.e., the total number of entity pairs involved in the head relation) in indicator design and only measures the number of entity pairs involved in the Horn rules. As the toy example shown in Figure 4, the number of Horn rules in the example is smaller than that of entity pairs with the same head relation in the rule. There is only one Horn rule, but the relation (i.e., *fatherOf*) involves four entity pairs. When measuring rule reliability, this paper only considers the triple (i.e., *father*(*C*, *B*)) involved in Horn rules and ignores the other three entity pairs of the relation (i.e., *father*).

Body				Head	
mother		marry		father	
A	B	A	C	C	B
		V	M	M	N
		Y	Q	Q	P
Z	F			E	F

Figure 4. The number of Horn rules is much lower than the number of head relationships.

Biased distribution of facts. The KGs are collections of triples extracted from the texts. The triples described in the text cannot be roughly uniformly distributed like datasets in other domains, such as signal processing. Even the standard dataset cannot guarantee the uniform distribution of the entities involved in each rule. The knowledge base constructed by the actual application cannot guarantee the balance even more. As shown in Figure 5, there may be different bodies pointing to the same head. In order to eliminate the imbalance of bodies in Horn rules, this paper only considers the number of head-to-tail pairs involved in the body. For instance, this paper only counts the types of head-to-tail pairs in Figure 5, i.e., the number of Horn rules is three.

Body				Head	
Live In		Belong To		Nationality	
A	California	California	America	A	America
A	Florida	Florida	America	A	America
B	Beijing	Beijing	China	B	China
C	Tokyo	Tokyo	Japan	C	Japan

Figure 5. There may be different bodies pointing to the same head.

Considering the uncertainty of the incomplete information and the biased distribution of facts, this paper designs a preliminary rule reliability index HRR_{ini} based on the Standard Confidence:

$$HRR_{ini} = \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y') : \vec{B} \wedge r(x, y') + \#(x', y) : \vec{B} \wedge r(x', y) - supp(\vec{B} \Rightarrow r(x, y))} \quad (9)$$

where $\#(x, y') : \vec{B} \wedge r(x, y')$ as an abbreviation for $|\{(x, y') : (x, y') \in \vec{B} \wedge r(x, y')\}|$, which refers to the number of all head-to-tail pairs that satisfy $r(x, ?)$, $x \in X$ (where X refers to the head entity that appears in the Horn rule $\vec{B} \Rightarrow r(x, y)$). $\#(x', y) : \vec{B} \wedge r(x', y)$ as an abbreviation for $|\{(x', y) : (x', y) \in \vec{B} \wedge r(x', y)\}|$, which refers to the number of all head-to-tail pairs that satisfy $r(?, y)$, $y \in Y$ (where Y refers to the tail entity that appears in the Horn rule $\vec{B} \Rightarrow r(x, y)$). The value of HRR_{ini} changes on the interval $(0, 1]$.

The credibility of rule reliability. In order to compare the relative reliability of Horn rules, the reliability index HRR_{ini} transforms the rules to the same scale. Nevertheless, the credibility of each rule's reliability is different. For example, if the Horn rules of the same relation have different occurrences, then the credibility of distinct Horn rules is different. The following formula shows that the first rule appears twice, and the reliability is 100%. The second rule appears 95 times, and its reliability is 95%. However, the second rule is more credible than the first rule.

$$HRR_{ini-1} = \frac{2}{2} = 100\% \quad (10)$$

$$HRR_{ini-2} = \frac{95}{100} = 95\% \quad (11)$$

In order to measure the credibility of the rule reliability, the model designs a credibility index: the ratio of the number of Horn rules and the number of head-to-tail pairs involved in the relation.

$$RC = \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y) : r(x, y)} \quad (12)$$

Horn rule reliability index. Combined with the credibility of the index, the final reliability index HRR is:

$$HRR = HRR_{ini} \times RC \quad (13)$$

This indicator is a comparison of different Horn rules for the same relation. The exact relation contains the same number of head-to-tail pairs. Therefore, by removing the denominator term, the reliability index of Horn rules can be simplified as follows.

$$HRR = \frac{(supp(\vec{B} \Rightarrow r(x, y)))^2}{\#(x, y') : \vec{B} \wedge r(x, y') + \#(x', y) : \vec{B} \wedge r(x', y) - supp(\vec{B} \Rightarrow r(x, y))} \quad (14)$$

In the first part of the HRER model, we calculate HRR of the searched Horn rule. For the various Horn rules from the same head, we rank the tails predicted by Horn rules according to HRR .

4.3. Reasoning Based on Entity Rules

Embedding-based methods are the process of learning the representations of entities and relations by solving an optimization problem of maximizing the scores of correct triples while minimizing the scores of error triples. The embeddings in KGs contain the links between relations and the connections between entities, e.g., the similarity of entities.

Limited representations restrict the performance of logic rules. Such methods only search for closed Horn rules, accounting for its poor performance in standard datasets. Mining richer representations of relation-rule are our future research route. Unlike embedding-based methods, logic rules do not consider the links between entities and only mine relation rules, which accounts for its poor performance. In order to alleviate this problem, this paper proposes entity rules.

The entity rule discussed in this paper refers to inclusion, i.e., a rule that an entity contains another entity. As shown in Figure 6, the rule that an entity contains another entity means the subordinate relationship between the entities. When two entities contain each other, then the two entities are equal. Given conditions that entity A belongs to entity B, we can infer that A has the same attributes as B. Entity rules achieve link prediction in this way.

Rule	Usage
$A \in B$	$B \Rightarrow A$
$C \in D, D \in C$	$C \Rightarrow D, D \Rightarrow C$

Figure 6. This is the inclusion of properties between entities.

Entity rule mining. Entity rule mining is similar to relation rule mining. This paper mines entity rules based on the association features of “pseudo triples”. The entity rules mined in this paper are similar to the single-hop Horn rules, so this paper utilizes the method of mining single-hop Horn rules to search for entity rules in “pseudo triples”.

We swap the tail and relation in the triple to reconstruct a new “pseudo triple”. See the example in Figure 7 for the generation of “pseudo triple”, i.e., the actual relation is regarded as “tail entity”, and the actual tail entity is regarded as “relationship”. In the first step of the rule mining, “pseudo triples” are input into the relationship rule mining program. We implement the mining of entity rules by searching for single-hop closed Horn rules.

	Head	Rel	Tail
True Triple	Team A	football_position/players	Player 1
Pseudo Triple	Team A	Player 1	football_position/players

Figure 7. This is an example for the conversion of triples to pseudo triples.

For the entities predicted by the entity rules, we rank them across the number of satisfied rules. Similar to the reliability of Horn rules HRR , we utilize the reliability of entity rules to sort the entity rules.

$$ERR = \frac{(supp(\vec{B} \Rightarrow t(h, r)))^2}{\#(h, r') : \vec{B} \wedge t(h, r') + \#(h', r) : \vec{B} \wedge t(h', r) - supp(\vec{B} \Rightarrow t(h, r))} \quad (15)$$

where h, r, t represent the heads, relation and tail in a triple, respectively. Where $\#(h, r') : \vec{B} \wedge t(h, r')$ is an abbreviation for $|\{(h, r') : (h, r') \in \vec{B} \wedge t(h, r')\}|$, which refers to the number of all head-to-tail pairs that satisfy $t(h, ?)$. $\#(h', r) : \vec{B} \wedge t(h', r)$ is an abbreviation for $|\{(h', r) : (h', r) \in \vec{B} \wedge t(h', r)\}|$, which refers to the number of all head-to-tail pairs that satisfy $r(?, y)$.

5. Experiments and Results

This section verifies the performance of HRER on link prediction tasks through experiments.

5.1. Datasets and Evaluations

Datasets for benchmarking link prediction should be obtained by sampling real-world KGs. We evaluate HRER using four standard link prediction datasets generated from actual scenarios (see Table 1). We can access four datasets through this link. (<https://github.com/ibalazevic/TuckER>, accessed on 20 January 2022).

- FB15k [1]. This dataset is a subset of Freebase, a large, growing knowledge base of the real world.
- FB15k-237 [37]. This dataset is obtained by eliminating the inverse and equal relations in FB15K, making it more difficult for simple models to do well.
- WN18 [1]. This dataset is a subset of WordNet, a hierarchical database containing lexical relations between words.
- WN18RR [27]. This dataset is achieved by excluding inverse and equal relations in WN18.

Table 1. Dataset statistics.

Dataset	#Entities	#Relations	#Triples	#Testset
FB15K	14,951	1345	483,142	59,071
FB15K-237	14,541	237	272,115	20,466
WN18	40,943	18	141,442	5000
Wn18RR	40,599	11	86,835	3134

Evaluation Settings. We use evaluation metrics standard across the link prediction literature: mean reciprocal rank (MRR) and Hits@ k , $k \in \{1, 3, 10\}$. Mean reciprocal rank is the average of the inverse of the mean rank assigned to the true triple overall candidate triples. Hits@ k measures the percentage of times a true triple is ranked within the top k candidate triples. We evaluate the performance of link prediction in the filtered setting [1], i.e., all known true triples are removed from the candidate set except for the current test triple. In both settings, higher MRR or higher Hits@1/3/10 indicate better performance.

5.2. Parameter Settings

HRER contains two parts: mining Horn rule and entity rule (see Figure 1). The AMIE algorithm uses parallel computing to accomplish rule mining, which significantly improves the efficiency of rule mining. Therefore, this paper applies the AMIE algorithm to mining rules, calculates the corresponding rule reliability HRR and ERR , and performs link prediction with the combination of two types of rules.

Unlike the embedding-based method with many parameter settings, HRER only sets one parameter in rule mining to control the number of rules. This paper only uses HRR between [0 and 1] in the rule mining step. Modifying this indicator will determine the number of searched rules. When we set the indicator to 0, all Horn rules are mined; if we take the value as 1, the algorithm will dig a few Horn rules. Generally speaking, the rule with lower rule reliability is less useful for link prediction. If the AMIE algorithm searches out all Horn rules, too many rules will affect the efficiency of link predictions. This paper sets HRR to 0.05 and ERR to 0.01 to control the number of rules.

5.3. Link Prediction Results

The experiments on link prediction mainly compare the following methods: TransE (i.e., the primary embedding method), STTransE, CrossE, TorusE, RotatE, TuckER, DisMult, ComplEx, ANALOGY, SimpleE, HolE, ConvE, ConvKB, ConvR, CapsE, RSN and AMIE. The parameters used by AMIE are the default parameters: Head Coverage equals 0.01, and PCA conf equals 0.1. The experiment first performs the link prediction for the first part and

the second part of the HRER model separately, then the two parts are merged to obtain the final result of HRER.

Result Comparison. As can be seen from Table 2, the performance of the Horn rules based on *HRR* designed in this paper exceed **PCA conf** in AMIE. Besides, entity rules obtain a relative improvement of 0.22% and 2.32% in MRR and Hits@10, averaged on FB15k, FB15k-237, WN18 and WN18RR. Overall, HRER outperforms previous state-of-the-art models on all metrics across two datasets (apart from FB15K-237, where Tucker does better).

HRER has only one parameter to control the number of Horn rules. If HRER does not set this parameter, performance on link prediction may be better. However, to reduce mining time by eliminating the redundancy of rules, HRER set the threshold to limit the number of rules. Since the model in this paper only mines closed Horn rules and simple entity rules, HRER is limited by the form of rule representation. Therefore, HRER cannot achieve the best Hit@10 on FB15K-237.

Case Study and Interpretability. The most significant feature of HRER is interpretability. For all prediction results of HRER, the model can provide the basis. Table 3 shows part of the reasoning basis for link prediction on FB15K-237.

As can be seen from Table 3, the reasoning basis is consistent with our perception. For example, we want to reason the releasing area of a specific movie A. It is known that the release area of A is Region C, and the adjacent region of Region C is Region B, so the model infers that the movie A will also be shown in Region B.

Table 2. Link prediction results on FB15k, FB15k-237, WN18 and WN18RR.

	FB15K			FB15K-237			WN18			Wn18RR		
	FHit@1/%	FHit@10/%	FMRR	FHit@1/%	FHit@10/%	FMRR	FHit@1/%	FHit@10/%	FMRR	FHit@1/%	FHit@10/%	FMRR
TransE	49.36	84.73	0.628	21.72	49.68	0.315	40.56	94.87	0.646	2.70	49.52	0.206
STransE	39.77	79.60	0.543	22.48	49.56	0.315	43.12	93.45	0.656	10.13	42.21	0.226
CrossE	60.08	86.23	0.702	21.21	47.05	0.298	73.28	95.03	0.834	38.07	44.99	0.405
TorusE	68.85	83.98	0.746	19.62	44.71	0.281	94.33	95.44	0.947	42.68	53.35	0.463
RotatE	73.93	88.10	0.791	23.83	53.06	0.336	94.30	96.0	0.949	42.80	57.15	0.476
DistMult	73.61	86.32	0.784	22.44	49.01	0.313	72.60	94.61	0.824	39.68	50.22	0.433
ComplEx	81.56	90.53	0.848	25.72	52.97	0.349	94.53	95.50	0.949	42.55	52.12	0.458
ANALOGY	65.59	83.74	0.726	12.59	35.38	0.202	92.61	94.42	0.934	35.82	38.00	0.366
Simple	66.13	83.63	0.726	10.03	34.35	0.179	93.25	94.58	0.938	38.27	42.65	0.398
HolE	75.85	86.78	0.800	21.37	47.64	0.303	93.11	94.94	0.938	40.28	48.79	0.432
Tucker	72.89	88.88	0.788	25.90	53.61	0.352	94.64	95.80	0.951	42.95	51.40	0.459
ConvE	59.46	84.94	0.688	21.90	47.62	0.305	93.89	95.68	0.945	38.99	50.75	0.427
ConvKB	11.44	40.83	0.211	13.98	41.46	0.230	52.89	94.89	0.70	95.63	52.50	0.249
ConvR	70.57	88.55	0.773	25.56	52.63	0.346	94.56	95.85	0.950	43.73	52.68	0.467
CapsE	1.93	21.78	0.087	7.34	35.60	0.160	84.55	95.08	0.890	33.69	55.98	0.415
RSN	72.34	87.01	0.777	19.84	44.44	0.280	91.23	95.10	0.928	34.59	48.34	0.395
AMIE	67.40	88.15	0.797	24.47	47.79	0.308	87.21	94.03	0.931	31.05	35.60	0.357
Horn Rule	84.27	89.01	0.861	25.10	48.22	0.312	93.47	95.32	0.941	44.16	50.98	0.465
Ent Rule	13.82	17.37	0.142	10.75	20.03	0.113	15.81	20.74	0.171	10.08	11.87	0.107
HRER	84.87	91.09	0.871	25.39	48.98	0.328	97.52	97.87	0.976	46.94	53.32	0.489

Table 3. Horn rules mined in FB237.

Dataset	#Entities	#Relations
Rule 1	head	(X, /sports/sports_team/roster./American_football/football_roster_position/position, Y)
	body	(X, /sports/sports_position/players./sports/sports_team_roster/team, Y)
Rule 2	head	(X, /award/award_category/winners./award/award_honor/ceremony/football_roster_position/position, Y)
	body	(X, /award/award_category/category_of, Z) (Z, /time/event/instance_of_recurring_event, Y)
Rule 3	head	(X, /film/film/release_date_s./film/film_regional_release_date/film_release_region, Y)
	body	(X, /film/film/release_date_s./film/film_regional_release_date/film_release_region, Z) (Z, /location/location/adjoin_s./location/adjoining_relationship/adjoins, Y)

6. Conclusions

This paper proposes a new bottom-up rule learning model for link prediction—HRER. The major novelty of HRER is as follows. First, HRER designs a new Horn rule filtering index *HRR* to measure the reliability of Horn rules. Furthermore, HRER proposes entity rules for the limitation of rule expression. In addition, HRER has better interpretability

and can give a better explanation for the inference. Finally, unlike the embedding-based method, HRER needs a minimal parameter to control the number of rules. Experiments on the standard dataset show that HRER achieves state-of-the-art performances. In the future, our research will no longer be restricted to closed logic rules, and we will study more representations of rules. Recently, graph neural networks have achieved good performance on link prediction. In the future, we also plan to leverage the graph attention framework to capture higher-order relations between entities.

Author Contributions: Conceptualization, Z.L. and J.Y.; validation, K.H.; formal analysis, Z.L.; investigation, H.L.; resources, L.C.; writing—original draft preparation, L.Q. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Anhui Provincial Natural Science Foundation OF FUNDER grant number No. 1908085MF202 and Independent Scientific Research Program of National University of Defense Science and Technology OF FUNDER grant number No. ZK18-03-14.

Informed Consent Statement: Not applicable.

Data Availability Statement: MDPI Research Data Policies at <https://github.com/ibalazevic/TuckER>, (accessed on 20 January 2022).

Acknowledgments: This work was partially supported by the Anhui Provincial Natural Science Foundation (No. 1908085MF202) and Independent Scientific Research Program of National University of Defense Science and Technology (No. ZK18-03-14).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NE, USA, 5–10 December 2013; pp. 2787–2795.
- Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735.
- Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E.R.; Mitchell, T.M. Toward an architecture for never-ending language learning. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010.
- Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A large ontology from wikipedia and wordnet. *J. Web Semant.* **2008**, *6*, 203–217. [\[CrossRef\]](#)
- Galárraga, L.A.; Teflioudi, C.; Hose, K.; Suchanek, F. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 413–422.
- Mahdy, A.; Lotfy, K.; Ismail, E.; El-Bary, A.; Ahmed, M.; El-Dahdouh, A. Analytical solutions of time-fractional heat order for a magneto-photothermal semiconductor medium with Thomson effects and initial stress. *Results Phys.* **2020**, *18*, 103174. [\[CrossRef\]](#)
- Mahdy, A.M. Numerical solutions for solving model time-fractional Fokker–Planck equation. *Numer. Methods Partial Differ. Equ.* **2021**, *37*, 1120–1135. [\[CrossRef\]](#)
- Gao, L.; Zhu, H.; Zhuo, H.H.; Xu, J. Dual Quaternion Embeddings for Link Prediction. *Appl. Sci.* **2021**, *11*, 5572. [\[CrossRef\]](#)
- Wang, P.; Zhou, J.; Liu, Y.; Zhou, X. TransET: Knowledge Graph Embedding with Entity Types. *Electronics* **2021**, *10*, 1407. [\[CrossRef\]](#)
- Wang, M.; Qiu, L.; Wang, X. A Survey on Knowledge Graph Embeddings for Link Prediction. *Symmetry* **2021**, *13*, 485. [\[CrossRef\]](#)
- Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 687–696.
- Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
- Balažević, I.; Allen, C.; Hospedales, T.M. Tucker: Tensor factorization for knowledge graph completion. *arXiv* **2019**, arXiv:1901.09590.
- Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197.
- Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of the ICML, New York, NY, USA, 20–22 June 2016.
- Goethals, B.; Van den Bussche, J. Relational association rules: Getting Warmer. In Proceedings of the Pattern Detection and Discovery, London, UK, 16–19 September 2002; pp. 125–139.

17. Schoenmackers, S.; Davis, J.; Etzioni, O.; Weld, D. Learning first-order horn clauses from web text. In Proceedings of the 2010 Conference on Empirical Methods on Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; pp. 1088–1098.
18. Meilicke, C.; Fink, M.; Wang, Y.; Ruffinelli, D.; Gemulla, R.; Stuckenschmidt, H. Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion. In Proceedings of the International Semantic Web Conference, Monterey, CA, USA, 8–12 October 2018; pp. 3–20.
19. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI, Quebec City, QC, Canada, 27–31 July 2014; Volume 14, pp. 1112–1119.
20. Zhang, Z.; Cai, J.; Zhang, Y.; Wang, J. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 3065–3072.
21. Maximilian, N.; Volker, T.; Hans-Peter, K. A Three-Way Model for Collective Learning on Multi-Relational Data. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, 28 June–2 July 2011; pp. 809–816.
22. Yang, B.; Yih, W.t.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv* **2014**, arXiv:1412.6575.
23. Liu, H.; Wu, Y.; Yang, Y. Analogical inference for multi-relational embeddings. *arXiv* **2017**, arXiv:1705.02426.
24. Kazemi, S.M.; Poole, D. Simple Embedding for Link Prediction in Knowledge Graphs. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018; pp. 4289–4300.
25. Nickel, M.; Rosasco, L.; Poggio, T.A. Holographic Embeddings of Knowledge Graphs. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Schuurmans, D., Wellman, M.P., Eds.; AAAI Press: Palo Alto, CA, USA, 2016; pp. 1955–1961.
26. Zhang, Y.; Yao, Q.; Dai, W.; Chen, L. AutoSF: Searching Scoring Functions for Knowledge Graph Embedding. In Proceedings of the 36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, 20–24 April 2020; pp. 433–444. [[CrossRef](#)]
27. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
28. Nguyen, D.Q.; Nguyen, T.; Nguyen, D.Q.; Phung, D.Q. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. *arXiv* **2018**, arXiv:1712.02121.
29. Nguyen, D.Q.; Vu, T.; Nguyen, T.; Nguyen, D.Q.; Phung, D.Q. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. *arXiv* **2019**, arXiv:1808.04122.
30. Vashishth, S.; Sanyal, S.; Nitin, V.; Talukdar, P. Composition-based Multi-Relational Graph Convolutional Networks. *arXiv* **2020**, arXiv:1911.03082.
31. Nathani, D.; Chauhan, J.; Sharma, C.; Kaul, M. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 4710–4723. [[CrossRef](#)]
32. Muggleton, S. Inverse entailment and Progol. *New Gener. Comput.* **1995**, *13*, 245–286. [[CrossRef](#)]
33. Tan, P.N.; Kumar, V.; Srivastava, J. Selecting the right interestingness measure for association patterns. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 32–41.
34. Meilicke, C.; Chekol, M.W.; Ruffinelli, D.; Stuckenschmidt, H. Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019; pp. 3137–3143.
35. Meilicke, C.; Chekol, M.W.; Fink, M.; Stuckenschmidt, H. Reinforced Anytime Bottom Up Rule Learning for Knowledge Graph Completion. *arXiv* **2020**, arXiv:2004.04412.
36. Galárraga, L.; Teflioudi, C.; Hose, K.; Suchanek, F.M. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB J.* **2015**, *24*, 707–730. [[CrossRef](#)]
37. Toutanova, K.; Chen, D.; Pantel, P.; Poon, H.; Choudhury, P.; 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, Lisbon, Portugal, 17–21 September 2015; pp. 1499–1509.