machine learning & knowledge extraction

MDPI

Article

Knowledgebra: An Algebraic Learning Framework for Knowledge Graph

Tong Yang [1,†], Yifei Wang [2,†], Long Sha [2,†], Jan Engelbrecht [1] and Pengyu Hong [2,*]

[1] Department of Physics, Boston College, Chestnut Hill, MA 02135, USA; yangto@bc.edu (T.Y.); jan@bc.edu (J.E.)
[2] Department of Computer Science, Brandeis University, Waltham, MA 02453, USA; yifeiwang@brandeis.edu (Y.W.); longsha@brandeis.edu (L.S.)
* Correspondence: hongpeng@brandeis.edu
† These authors contributed equally to this work.

Abstract: Knowledge graph (KG) representation learning aims to encode entities and relations into dense continuous vector spaces such that knowledge contained in a dataset could be consistently represented. Dense embeddings trained from KG datasets benefit a variety of downstream tasks such as KG completion and link prediction. However, existing KG embedding methods fell short to provide a systematic solution for the global consistency of knowledge representation. We developed a mathematical language for KG based on an observation of their inherent algebraic structure, which we termed as *Knowledgebra*. By analyzing five distinct algebraic properties, we proved that the semigroup is the most reasonable algebraic structure for the relation embedding of a general knowledge graph. We implemented an instantiation model, *SemE*, using simple matrix semigroups, which exhibits state-of-the-art performance on standard datasets. Moreover, we proposed a regularization-based method to integrate chain-like logic rules derived from human knowledge into embedding training, which further demonstrates the power of the developed language. As far as we know, by applying abstract algebra in statistical learning, this work develops the first formal language for general knowledge graphs, and also sheds light on the problem of neural-symbolic integration from an algebraic perspective.

Keywords: algebraic learning; knowledge graph; category; semigroup; logic reasoning; neural-symbolic integration

Citation: Yang, T.; Wang, Y.; Sha, L.; Engelbrecht, J.; Hong, P. Knowledgebra: An Algebraic Learning Framework for Knowledge Graph. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 432–445. https://doi.org/10.3390/make4020019

Academic Editor: Isaac Triguero

Received: 26 March 2022
Accepted: 3 May 2022
Published: 5 May 2022

1. Introduction

Knowledge graphs (KGs) has raised enormous attention among the general artificial intelligence community, which represent human knowledge as a triplet data structure (*head entity*, *relation*, *tail entity*) and can be applied in various downstream scenarios, such as recommendation system [1], question answering [2–4], information extraction [5,6], and etc. [7–9]. It is therefore important to design appropriate knowledge graph embeddings (KGEs) to capture knowledge in the whole dataset with uniform consistency. It is therefore important to design appropriate knowledge graph embeddings (KGEs) to capture knowledge in the whole dataset with uniform consistency. A knowledge graph represents a network of real-world entities—i.e., objects, events, situations, or concepts—and illustrates the relationship between them. It is usually represented as one collection of triplets, where each triplet represents the relation between two entities. Figure 1 provides an illustration of a triplet (A, B, C) where A and C represent two entities while B is the relation between them. Triplet instances could be (Louver, is_located_in, Paris) and (Da Vinci, painted, Mona Lisa). Concretely, KGs are collections of factual triplets, where each triplet represents the relation between two entities [10,11]. Mathematically, a KG consists of two sets: an entity

set $\mathcal{E} = \{e_i\}_{i=1}^{N_e}$ and a relation set $\mathcal{R} = \{r_j\}_{j=1}^{N_r}$. Knowledge is represented as atomic triplets in the following form:

$$(e_i, r, e_j), \tag{1}$$

which can be interpreted as the following: the entity e_i is of the relation e to the entity e_j.

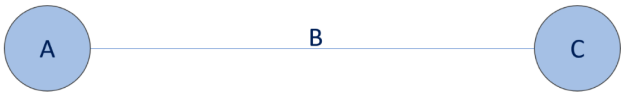

Figure 1. One triplet in KG, where A and C represent two entities while B is the relation between them. Relation B is usually directional.

KGE aims to encode entities and relations of triplet (e_i, r, e_j) into a continuous vector space, i.e., (e_i, r, e_j), associated with an operation $O_{\mathbf{R}}(\cdot)$ that ideally maps e_i to e_j (in the current work, we use regular letters to represent the semantic context of entities and relations, and bold letters to represent the high-dimensional array embedding of them). Quantitatively, the performance of an embedding model could be roughly measured by the distance between the mapping result, $O[e_i, r]$, and the tail entity embedding, e_j, which is based on a given metric, $\mathcal{D}_E$, defined in the entity embedding space. The representation design of a single triplet is trivial, while the challenge of KGE lies in the fact that different triplets share entities and relations, which requires a uniform representation of all elements that could consistently represent all triplets in a dataset. With the **TransE** model [12] as the starting baseline, researchers have explored the problem of KGE in majorly three directions:

Previous works [13–16] implemented different metrics $\mathcal{D}_E$, which control the efficiency of the entity embedding, where both euclidean based similarity scores and cosine similarity are among the top popular choices. The other set of preceding works in [17–19] instead designed various operations $O[\cdot, \cdot]$, which determine the consistency of the relation embedding. The attempted operations include simple ones such as vector addition, and vector multiplication, to complicated neural network-based operations including convolution and recurrent structures. Another interesting category of works combines KG embedding and logical rules using rule-based regularization or probabilistic model approximation, which can be found in [20–25]. All of the directions have rich mathematical structures. However, a formal analysis from the perspective of mathematics has been lacking for the general KGE tasks, which leaves the modeling design ungrounded.

In this work, we target the second direction, i.e., consistent relation embedding, and develop a formal language for a general KGE problem. okSpecifically, we observe that the consistency issue in relation embeddings directly leads to an algebraic description, and therefore offers an abstract algebra framework for KGE, which we termed as ***Knowledgebra***. The explicit structure of the Knowledgebra is determined by the details of a specific task/dataset, which could differ in five properties: *totality, associativity, identity, invertibility, and commutativity*. Regarding the rational behind the choice of five properties, we aim to investigate the general properties of relations in KG and the choice of five is a summary of previous works. For instance, previous studies [10,26] have considered certain specific inter-relation types including (anti)-symmetry, inversion, and compositional relations, while [27] makes extensions. We notice that relations in a general KG should be embedded in a semigroup structure, and hence propose a new embedding model, **SemE**, which embeds relations as semigroup elements and entities as points in the group action space. Furthermore, within the language of Knowledgebra, human knowledge about relations (also called logic rules in the following part) could be expressed by relation compositions. We, therefore, propose a simple method to directly integrate logic rules of relations with fact triplets to obtain better embedding models with improved performance. This method also provides a data-efficient solution for tasks with fewer training fact triplets but with a rich domain knowledge.

Our work is partially motivated by **NagE** [27], but differs from it significantly in the following aspects. Firstly, we deliver a categorical language for KGE problems, which is much more general than NagE with fewer assumptions; secondly, we prove that a group structure would be inappropriate for a large class of problems, where the invertibility could not be enforced; thirdly, beyond a conventional KGE perspective, we adopt a machine reasoning perspective by considering the impact of chain-like logic rules, which is traditionally studied in symbolic AI, and therefore shed light on a potential pathway for neural-symbolic integration.

The rest part of the paper is organized in the following ways: Section 2 introduces the emergent algebra in KG, i.e., Knowledgebra, and proves that a semigroup structure is suited for a general KGE task; Section 3 proposes a model, **SemE**, for general KGE problems, as an instantiation of Knowledgebra, and demonstrates its performance advantage on benchmark datasets; in Section 4, we propose a regularization based method to integrate chain-like logic rules into embedding model training, and deliver a case study using a toy dataset where logic rules are easy to be specified; in the end, we provide a further investigation on the implementation of **SemE** and discuss potential directions in the future in Section 5, which could exploit more power of the developed algebraic language, Knowledgebra, in knowledge graph applications.

2. Knowledgebra: An Emergent Algebra in Knowledge Graph

In this section, we would analyze a general KG, and demonstrate the emergence of an algebraic structure, which we term as *Knowledgebra*. The study of algebraic properties in Knowledgebra would produce constraints on KGE modeling.

2.1. A Categorical Language for Knowledge Graph

As introduced at the beginning, KGs are composed of two sets: $\mathcal{E}$ and $\mathcal{R}$, with entities in $\mathcal{E}$ linked by arrows representing relations in $\mathcal{R}$. Although knowledge triplets $\{(e_i, r, e_j)\}$ are the elementary atomic components of a KG, the complexity of the KG structure is not present on the triplet level. It is the set of logic rules that dictate the global consistency of a KG. Logic rules are the central topic of machine reasoning. In the machine reasoning field, relations are a special type of predicates, labeled as α, with arity 2. A logic rule can be expressed as the following:

$$\alpha_0 \leftarrow (\alpha_1, \alpha_2, \cdots, \alpha_m), \tag{2}$$

where each α_i is a predicate with entity variables as arguments. The above expression means that the head predicate α_0 would be iff all body predicates $\{\alpha_i\}_{i=1}^m$ hold. There is a special type of logic rules, chain-like rules, which has the following form:

$$r_0[e_1, e_{m+1}] \leftarrow (r_1[e_1, e_2], r_2[e_2, e_3], \cdots, r_m[e_m, e_{m+1}]), \tag{3}$$

where all predicates are of arity 2, and the head argument of the next predicate is always the tail argument of the previous one. The "cancellation" of intermediate terms $\{e_i\}_{i=2}^m$ implies a compositional definition of the corresponding type of logic rules, where a *composition* of two predicates r_a and r_b is denoted as $r_a \circ r_b$. Furthermore, it has been proved in [27] that the composition defined above is associative.

The chain rule reflects more complex logic rules, i.e., hyper-relations in KG. This is the central topic of machine reasoning since the model should learn to make inference via integrating information from multiple triplets. For example, the reasoning of "James visited Paris" could be completed from two triplets (James, visited, Tour Eiffel) and (Tour Eiffel, is located in, Paris). Here the chain rule becomes:

$$\textbf{visited}[\text{James, Paris}] \leftarrow (\textbf{visited}[\text{James, Tour Eiffel}], \textbf{isLocatedIn}[\text{Tour Eiffel, Paris}]). \tag{4}$$

Thus the chain-like reasoning from different levels of locations can not be ignored.

All elements discussed above have indicated the existence of an abstract mathematical structure: *category*. In mathematics, a category C consists of [28]:

- A class $ob(C)$ of objects;
- A class $hom(C)$ of morphisms, or arrows, or maps between the objects;
- A domain, or source object class function $dom : hom(C) \to ob(C)$;
- A codomain, or target object class function $cod : hom(C) \to ob(C)$;
- For every three objects a, b, and c, a binary operation $hom(a, b) \times hom(b, c) \to hom(a, c)$ called composition of morphisms; the composition of $f : a \to b$, and $g : b \to c$ is written as $g \circ f$;

such that the following axioms hold:

1. **Associativity:** if $f : a \to b$, $g : b \to c$ and $h : c \to d$, then $h \circ (g \circ f) = (h \circ g) \circ f$;
2. **Identity:** for every object x, there exists a morphism $1_x : x \to x$, called the identity morphism for x, such that every morphism $f : a \to x$ satisfies $1_x \circ f = f$, and every morphism $g : x \to b$ satisfies $g \circ 1_x = g$.

It is straightforward to examine that all above definitions and axioms hold for a general knowledge graph, which therefore suggests that knowledge graphs naturally host a categorical language description. In this work, all our later discussions would then utilize concepts and properties of categories, which provide a formal basis.

2.2. Logic Construction versus Logic Extraction

With regards to logic rules, there are two pathways in the research of knowledge graphs: namely, *logic construction* and *logic extraction*.

Logic construction is widely used in machine reasoning via symbolic programming, where predicates are built as modules and logic rules are constructed explicitly. This is similar to the case of a theorem prover, where the propagation from sub-queries to a query is governed by pre-defined rules composed of logical operators, e.g., conjunctions and disjunctions. Logic construction is a completely deductive process. With explicit logic construction, one could derive both conclusions and reasoning paths at the same time. There are several advantages of applying logic construction: firstly, one could integrate common sense knowledge and domain expertise into the modeling of the reasoning process, which requires less or nearly zero data dependence; secondly, as rules are constructed explicitly, rigorousness could be guaranteed; thirdly, with the potential to construct a complete reasoning path, the interpretability of a conclusion derivation could be easily achieved. On the other side, the disadvantages of logic construction-based approaches are also significant:

1. The hand-crafting effort of integrating logic rules becomes impractical when the number of rules gets large;
2. The explicit construction could not accommodate any possible faults;
3. The construction could only take into account rules known a priori, and could not observe new ones (with logic operators, higher-order rules could be composed; However, here we refer to an inductive process to obtain new elementary rules).

These problems have been addressed in an alternative method: logic extraction.

Different from logic construction, logic extraction-based approaches belong to the category of statistical learning. Opposite to the spirit of logic construction, logic extraction is an inductive process, which infers that logic rules are implied by a collection of data samples. One of the most important advantages of logic extraction is that the human effort remains low when the number of logic rules increases, while logic construction needs to construct each rule one by one manually. Thus, logic extraction could take advantage of huge datasets and is fault-tolerant based on its statistical nature. Besides, the induction process is insensitive to the number of logic rules and hence scales well with an increasing number of rules. It is obvious, though, that logic extraction could not integrate with human knowledge easily, and also suffers from the interpretability issue.

It is also noteworthy to emphasize an extra challenge for logic extraction on the implementation level: the set of logic rules hidden in a dataset requires a *global consistency* of knowledge representation. In the context of KGE, relation embeddings are not independent of each other and should accommodate all chain-like logic rules under compositions.

2.3. Algebraic Constraints in KGE

With the discussion above, we now consider the problem of KGE. KGE belongs to the class of logic extraction which explores a given dataset to infer logic rules. There are two embeddings of KGE, i.e., entity embeddings and relation embeddings. The chain-like logic rules, however, are entity-independent and only related to relation embeddings. As introduced above, relations correspond to morphisms in a category and are not independent of each other due to the existence of hidden logic rules under compositions.

The class $hom(C)$, i.e., the set of relations, forms an algebraic structure, which, in general, is termed as *knowledgebra*. To specify an algebraic structure, the following five properties are usually discussed:

- **Totality:** $\forall r_a, r_b \in hom(C), r_a \circ r_b$ is also in $hom(C)$;
- **Associativity:** $\forall r_a, r_b, r_c \in hom(C), r_a \circ (r_b \circ r_c) = (r_a \circ r_b) \circ r_c$;
- **Identity:** $\exists e \in hom(C), \forall r \in hom(C), e \circ r = r \circ e = r$;
- **Invertibility:** $\forall r \in hom(C), \exists \bar{r} \in hom(C), r \circ \bar{r} = \bar{r} \circ r = e$;
- **Commutativity:** $\forall r_a, r_b \in hom(C), r_a \circ r_b = r_b \circ r_a$.

Variant algebraic structures could be differed by these five properties, and we list 10 well-studied structures in Table 1.

Table 1. Various algebraic structure and their properties [29].

	Totality	**Associativity**	**Identity**	**Invertibility**	**Commutativity**
semigroupoid	-	✓	-	-	-
small category	-	✓	✓	-	-
groupoid	-	✓	✓	✓	-
magma	✓	-	-	-	-
unital magma	✓	-	✓	-	-
loop	✓	-	✓	✓	-
semigroup	✓	✓	-	-	-
monoid	✓	✓	✓	-	-
group	✓	✓	✓	✓	-
abelian group	✓	✓	✓	✓	✓

To fully specify the structure of Knowledgebra, we now examine the five properties in the context of KGE. An analysis in [27] claimed that all the first four properties: totality, associativity, identity, and invertibility, should hold for KGE modeling, and the authors thus developed a group-based framework for relation embeddings. While we agree with most of the analysis in [27], we now provide an argument specifically on the invertibility property. Consider the following logic rule example consisting of two kinship relations:

$$r_a = \textbf{isMotherOf}, \quad r_b = \textbf{isBrotherOf}, \\ r_a \circ r_b = r_a. \tag{5}$$

Now if a group structure is used for relation embedding, then there always exist an inverse relation $\bar{r}_a$ for r_a, then, based on associativity, we would obtain:

$$r_b = (\bar{r}_a \circ r_a) \circ r_b = \bar{r}_a \circ (r_a \circ r_b) = \bar{r}_a \circ r_a = e, \tag{6}$$

requiring the relation **isBrotherOf** to be an identity map that always returns the head entity itself—which is incorrect. Therefore the existence of r_a should be prohibited. Another less trivial example consists of the following four kinship relations:

$$r_a = \textbf{isSonOf}, \quad r_b = \textbf{isMotherOf}, \quad r_c = \textbf{isFatherOf}, \quad r_d = \textbf{isBrotherOf}, \tag{7}$$

which could be related by the following two rules abstractly:

$$r_a \circ r_b = r_d, \tag{8}$$
$$r_a \circ r_c = r_d. \tag{9}$$

Again, if a group embedding is implemented, based on invertibility, i.e., $\bar{r}_a$, and associativity, we would obtain:

$$r_b = (\bar{r}_a \circ r_a) \circ r_b = \bar{r}_a \circ (r_a \circ r_b) = \bar{r}_a \circ r_d, \tag{10}$$
$$r_c = (\bar{r}_a \circ r_a) \circ r_c = \bar{r}_a \circ (r_a \circ r_c) = \bar{r}_a \circ r_d, \tag{11}$$

which then demands directly:

$$r_b = r_c, \tag{12}$$

an obviously incorrect conclusion. To simultaneously accommodate the two equations in Equation (8), the element r_a should not be invertible. This suggests that invertibility is not a desired property for relation embedding in KG. As in [27], the existence of an identity element is proved based on invertibility, which we could also ignore for now (the existence of identity is not necessary but indeed compatible without any conflict). Therefore, in the end, only totality and associativity are natural properties of KGE tasks, which, according to Table 1, indicates that a semigroup-based relation embedding is desired.

3. A Semigroup Based Instantiation of Knowledge Graph Embedding

In the above section, we delivered a formal analysis of KGE problems and proved that relations in a KG could generally be embedded in a semigroup structure. In this section, we implement this proposal by constructing an instantiation model, termed as **SemE**, and demonstrate the power of algebraic-based embedding on several benchmark tasks.

3.1. Model Design and Analysis

We firstly introduce the proposed model, including embedding space and distance function design.

3.1.1. Embedding Spaces for Entities and Relations

We choose the simplest semigroup, which has a straightforward parametrization: real $k \times k$ matrices, as the embedding space for relations. It reduces to $GL(k, \mathbb{R})$ with an extra condition: $det \neq 0$, which guarantees the invertibility. Here $GL(k, \mathbb{R})$ represents the general linear group, which is the set of k-by-k invertible matrices over real numbers $\mathbb{R}$, together with the operation of matrix multiplication. Entities are embedded as real vectors. Similar to the implementation in [27], to prevent the curse of dimensionality while allowing an embedding

space large enough to accommodate knowledge graphs, we apply block-diagonal matrices as relation embeddings:

$$\left[\begin{array}{c|c|c|c} M_1 & 0 & \dots & 0 \\ \hline 0 & M_2 & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & M_n \end{array}\right] \left[\begin{array}{c} v_1 \\ \hline v_2 \\ \hline \vdots \\ \hline v_n \end{array}\right], \tag{13}$$

where each M_i is a real $k \times k$ matrix and each v_i is a vector in $\mathbb{R}^k$, i.e., entities are embedded in $(\mathbb{R}^k)^{\otimes n}$. We label the $(nk) \times (nk)$ embedding matrix for relation r as M_r, and the (nk)-dim embedding vector for entity e as v_e.

3.1.2. Distance Function for Similarity Measure

To apply an end-to-end gradient-based training, a scoring function to compare the similarity between two arbitrary entities is required. In this work, we exploited a distance-based scoring function that measures the plausibility of a factual triplet as the distance between the two entities, where a translation of the head entity is usually carried out by the relation. The two most common choices are Euclidean distance and cosine distance. The latter one, i.e., cosine similarity, focuses only on the relative angle between two high dimensional vectors while ignoring the radial component. Ref. [30] overviews more scoring function options. In the current work, a general $k \times k$ matrix transform a k-dim vector in 6 ways, including 5 affine-type transformations: translations, rotations, reflections, scaling maps, and shear maps, and projections achieved by non-invertible matrices, most of which, except rotations and reflections, cannot be differed by the cosine similarity, and we, therefore, choose Euclidean distance to measure entity similarity. For a fact triplet (e_i, r, e_j), the performance of a SemE model would therefore be measured by the following similarity measure:

$$s_r(e_i, e_j) = \|M_r v_{e_i} - v_{e_j}\|_2, \tag{14}$$

where $\|\cdot\|_2$ calculates the L_2-norm of a vector.

The complete loss function is designed as follows:

$$\begin{aligned} \mathcal{L}_0 = & -\frac{1}{p_{\text{loss}}+1}(\log \sigma[\gamma - s_r(e_i, e_j)] \\ & + p_{\text{loss}} \sum_{k=1}^{n} p(e'_{ik}, r, e'_{jk}) \log \sigma[s_r(e'_{ik}, e'_{jk}) - \gamma]) \\ p\left(e'_{ik}, r, e'_{jk} | \{e_i, r, e_j\}\right) = & \frac{e^{\alpha[\gamma - s_r(e'_{ik}, e'_{jk})]}}{\sum_l e^{\alpha[\gamma - s_r(e'_{il}, e'_{jl})]}} \end{aligned} \tag{15}$$

where σ is the Sigmoid function, and γ is a hyper-parameter controlling the margin to prevent over-fitting, p_{loss} is a hyper-parameter controlling the ratio of negative and positive losses. Equation (15) is the standard form that was first proposed in [12], and applied in nearly all KGE models, including [10–12,26,27,31,32], etc. Following an energy-based

framework, the energy of a triplet is equal to the similarity measure of this triplet, which corresponds to Equation (14). To learn embeddings, a margin-based ranking criterion over the training set is proposed, which prefers ranking real triplets over corrupted triplets. Specifically, the set of corrupted triplets is constructed from negative sampling and is composed of training triplets with either the head or tail (but not both), replaced by a random entity. The complete loss favors lower values of the energy for training triplets than for corrupted triplets and thus leads to the two components in Equation (15). We apply a popularly implemented [10,27] negative sampling setup, termed as *self-adversarial negative sampling* [10], with e'_{ik} and e'_{jk} being the negative samples for head and tail entity, respectively, while $p(e'_{ik}, r, e'_{jk})$ is the adversarial sampling weight with the inverse temperature α controlling the focus on poorly learned samples.

3.1.3. Low Dimensional Relation Embedding

In the standard SemE, relations are embedded as n blocks of $k \times k$ matrices while the entities are mapped to (nk)-dim vectors. In practice, there are tasks where only simple relations are involved. For example, the WordNet-18 dataset includes only 18 distinct relations, connected by very simple logic rules. However, the large number of entities requires a relatively high dimensional vector space for embedding, which easily results in large redundancy in relation embedding in such tasks. To improve the efficiency of parametrization for tasks with simple relations, and accelerate learning convergence at the same time, we propose two simplified alternatives for relation embedding:

- **shared blocks:** instead of using n distinct $k \times k$ matrices, we use identical copies of one $k \times k$ matrix, i.e., $M_i = M_0, \forall i \in [1, n]$. The number of parameters of embedding for one relation then reduces from $n \times k \times k$ to $k \times k$, which is a super low dimensional embedding, termed as **SemE-s**.
- **shared blocks with shift:** in the case where a single $k \times k$ matrix is insufficient while low-dimensional efficiency is still demanded, we could break the symmetry among n subspaces by introducing a block-dependent shift δ_i. Precisely, the transformation in each subspace could be written as:

$$M_0 \cdot v_i + \delta_i, \quad \forall i \in [1, n]. \tag{16}$$

  The number of parameters is then $k \times k + n \times k$. And we term the resulting model as **SemE-δs** (importantly, this shift corresponds to a translation in each subspace, which, together with the matrix multiplication, still hold a semigroup structure. The resulting operation is quite similar to a Euclidean group but with non-invertible elements).

We would implement these low-dim embedding methods later on tasks with simple relations.

3.2. Experiments on Benchmark Datasets

3.2.1. Experimental Setup

Datasets: we evaluate the proposed approach on two popular public knowledge graph benchmarks: WN18RR [33] and FB15k-237 [34]. These two datasets were derived from WN18 [35] and FB15K [36] respectively. The FB15k dataset extracted all FreeBase entities that have over 100 mentions and are featured in the Wikilinks database while the WN18 dataset extracted from a linguistic knowledge graph ontology named the WordNet. After finding that the FB15k and the WN18 dataset suffered from test leakage issues due to the presence of equivalent inverse relations, the WN18RR and FB15k-237 were created as more challenging datasets, removing all equivalent and inverse relations. These two datasets are currently benchmarked across the KGE domain to fairly compare model performances specifically in recent relevant works [11,27,34]. In these two datasets, none of the triplets in the training set are directly linked to the validation and test sets.

Evaluation Metrics: similar to previous work, we use two ranking-based metrics for evaluation: (1) Cut-off Hit ratio (H@N, $N \in \{1, 3, 10\}$), which measures the proportion of correct entity predictions among the top N prediction result cut-off, and (2) Mean Reciprocal Rank (MRR), which represents the average of inverse ranks assigned to correct entities.

Implementation Details: we implement our models via the Pytorch framework and experimented on a server with an NVIDIA Tesla V100 GPU (32 GB). The Adam optimizer [37] is used with default settings of β_1 and β_2. We use a learning rate annealing schedule that discounted the learning rate by a factor of 0.1 with a patience setting of 10. The batch size is fixed at 1000 (the code is available at https://github.com/yifeiwang15/Knowledgebra (accessed on 2 May 2022)).

3.2.2. Experiment Results

For FB15k-237, we implement the standard **SemE** as stated in Section 3.1.1, with parameterization of $k = 5$ and $n = 240$. Other hyper-parameters are tuned as following: learning rate $\eta \in \{3e-4, 1e-3\}$; number of negative samples during training $n_{\text{neg}} \in \{64, 128\}$; adversarial negative sampling temperature $\alpha \in \{0.75, 0.85, 0.95, 1\}$; loss function margin $\gamma \in \{9, 12\}$; ratio between negative and positive losses $p_{\text{loss}} \in \{5, 10\}$. The best model is under configuration of $\eta = 1e-3$, $n_{\text{neg}} = 64, \alpha = 0.85, \gamma = 9, p_{\text{loss}} = 5$. As another benchmark dataset, WN18RR only includes simple relations that can be sufficiently captured by low-dimensional embeddings. Therefore we apply a low-dim alternative model, **SemE-δs**, as discussed in Section 3.1.3, where we take $k = 10$ and $n = 100$ in this case. Other hyper-parameters of grid search include: $\eta \in \{3e-4, 1e-3\}$; $n_{\text{neg}} \in \{64, 128\}$; $\alpha \in \{0.5, 0.7, 0.85, 1\}$; $\gamma \in \{6, 7, 7.5\}$; $p_{\text{loss}} \in \{10, 20, 30\}$. The best performance appears in the configuration with $\eta = 1e-3$, $n_{\text{neg}} = 128, \alpha = 0.7, \gamma = 6, p_{\text{loss}} = 30$. The experimental results of the best models are exhibited in Table 2.

Table 2. Experiment results on WN18RR and FB15k-237 datasets (best scores are marked as bold while the second best are underlined). A standard **SemE** model is applied for FB15k-237, while a low-dim alternative **SemE-δs** is used for WN18RR.

Model	WN18RR				FB15k-237			
	MRR	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
TransE [12]	0.226	-	-	0.501	0.294	-	-	0.465
ComplEx [38]	0.440	0.410	0.460	0.510	0.247	0.158	0.275	0.428
DistMult [39]	0.430	0.390	0.440	0.490	0.241	0.155	0.263	0.419
ConvE [33]	0.430	0.400	0.440	0.520	0.325	0.237	0.356	0.501
MuRE [1] [32]	0.475	<u>0.436</u>	0.487	0.554	0.336	0.245	0.370	0.521
RotatE [10]	0.476	0.428	0.492	<u>0.571</u>	0.338	0.241	0.375	<u>0.533</u>
NagE [27]	<u>0.477</u>	0.432	<u>0.493</u>	**0.574**	<u>0.340</u>	<u>0.244</u>	<u>0.378</u>	0.530
SemE	**0.481**	**0.437**	**0.499**	0.567	**0.354**	**0.258**	**0.393**	**0.548**

[1] This is the Euclidean analogue of MuRP [32].

As shown above, in the WN18RR dataset, our SemE model outperformed the previous state-of-the-art knowledge graph model on the metrics of the average of inverse ranks assigned to correct entities, also known as the mean reciprocal rank, the cut-off hit ratio of top one and top three; on the FB15k-237 dataset our SemE model outperformed all the benchmark evaluation metrics compared to previous state-of-the-art model, our cut-off hit ratio at top three outperformed by a margin of 4%. Remarkably, in the task of WN18RR, our model SemE has already provided promising results only with a dimensionality setting of $k = 10$ and $n = 100$. In comparison, the baseline RotatE model has 81% more model parameters. A significantly small number of model parameters further demonstrates the advantage of the proposed approach.

4. Integrating Human Knowledge into Knowledge Graph Embedding

We have discussed the advantages and shortages of the logic constructions versus logic extraction in Section 2.2. In this section, we will propose a way to integrate human knowledge into KGE. This is valuable since logic rules, e.g., chain-like rules, could provide rich information and hence efficient constraints on the embedding model, which has been ignored in nearly all preceding works. With a regularization-based method to integrate chain-like logic rules derived from human knowledge into embedding training, we provided a solution to bridge the gap between logic construction and extraction. The resulting method is therefore more data-efficient and interpretable.

4.1. A Regularization Method for Logic Rules

One of the major challenges in KGE is to integrate human knowledge, either commonsense or domain knowledge, into the embedding model. As introduced in Section 2.2, knowledge is expressed as logic rules, which in turn is represented by relation compositions. The task of integrating human knowledge is equivalent to enforcing the compositional dependence among embeddings of different relations. For example, the two relations $r_a =$ **isWifeOf** and $r_b =$ **isHusbandOf** are mutually dependent on each other as:

$$r_a \circ r_b = E, \tag{17}$$

where E is an identity mapping. When a matrix embedding is implemented, the following equation should hold:

$$M_{r_a} \cdot M_{r_b} = \mathbb{I}, \tag{18}$$

which results into an identity matrix. The above example inspires a way to integrate human knowledge, i.e., logic rules, into embeddings: to design an additional loss term that minimizes the matrix distance suggested by rules. For the instance above, we may add the following term into loss function:

$$\mathcal{L} = \mathcal{L}_0 + \lambda \| M_{r_a} \cdot M_{r_b} - \mathbb{I} \|_2, \tag{19}$$

where $\mathcal{L}_0$ is the usual training loss defined in Equation (15), while the second term regularizes the embeddings of r_a and r_b to be mutually dependent. In general, for chain-like rules, which could be captured as compositions, we could apply the following regularized loss function:

$$\mathcal{L} = \mathcal{L}_0 + \sum_{i=1}^{K} \lambda_i \| M_{i,1} \cdot M_{i,2} - M_{i,3} \|_2, \tag{20}$$

where without loss of generality as argued in the *inverse and compositional hyper-relation*, logic rules are expressed as a compositional dependency of three relations, with one of which could be an identity to capture the case of the inverse. This provides an efficient approach to integrating human knowledge, i.e., logic rules, into KG embedding tasks.

4.2. Kinship: A Case Study of Logic Integration

We now demonstrate the above proposed regularized loss method on a toy dataset: Hinton's *Kinship* dataset. There are 12 relations in this toy KG: *wife*, *husband*, *father*, *mother*, *son*, *daughter*, *sister*, *brother*, *uncle*, *aunt*, *niece*, and *nephew*.

From common sense knowledge, we consider the following set of constraints for relation embedding shown in Table 3.

Table 3. Common sense knowledge in KINSHIP.

r_a	r_b	$r_a \circ r_b$
isWifeOf	**isHusbandOf**	**Identity**
isHusbandOf	**isMotherOf**	**isFatherOf**
isSonOf	**isMotherOf**	**isBrotherOf**
isSonOf	**isFatherOf**	**isBrotherOf**
isBrotherOf	**isFatherOf**	**isUncleOf**
isBrotherOf	**isMotherOf**	**isUncleOf**
isSisterOf	**isFatherOf**	**isAuntOf**
isSisterOf	**isMotherOf**	**isAuntOf**
isSonOf	**isBrotherOf**	**isNieceOf**

We implemented the logic regularized loss method using a small shared block model, **SemE-s**, with 2 copies of 2 dimensional subspaces. We used the batch size of 5 for training and 12 for testing. For other hyper-parameters we took $\alpha = 0.1, n_{\text{neg}} = 4, p_{\text{loss}} = 2$. We set all regularization parameters $\lambda_i = 0.1,\ \forall i$, and compared the baseline model with $\lambda_i = 0,\ \forall i$. Experimental results on testing dataset are shown in Table 4.

Table 4. Testing results on **SemE-s** model and logic regularized **SemE-s** model.

Model	MR	MRR	*H*@1	*H*@3	*H*@10
SemE-s	4.83	0.464	0.292	0.458	0.875
regularized **SemE-s**	**3.71**	**0.574**	**0.458**	**0.583**	**0.958**

The performance advantage of the logic regularized model is significant, which demonstrates the power of the regularization brought by logic rules. This showcases an efficient way to integrate external knowledge.

5. Discussion

SemE applies matrices to embed relations. A non-invertible matrix $\mathbf{M}$ has a determinant $det(\mathbf{M}) = 0$, which, from a dimensional-perspective, suggests a projection associated with a dimension reduction. For the example in Equation (7), the relation $r_a =$ **IsMotherOf** should not be invertible for a family with multiple children, since all vectors corresponding to a *child* should be simultaneously mapped by the matrix M_{r_a} to the same vector, which represents their mother. In other words, the non-invertible elements in a semigroup are used to capture N-to-1 relations, which commonly exist in real-life datasets.

A derived question from the above discussion would be the representation of 1-to-N relations. Within the context of KGE using statistical learning-based representation, it is challenging to directly design a mathematically rigorous 1-to-N mapping operation $O(\cdot, \cdot)$, as it always produces a deterministic result. However, this could be relieved by noting that the final performance of an embedding model is determined not directly by the mapping output but by the ranking of closeness between the output with each candidate entity. Therefore, instead of producing multiple results, the distributed learning framework requires the output to be as equidistant to all correct candidates as possible. This also explains the necessity of high-dimensional entity embedding: within a low-dimensional vector space, it is more challenging to find a point equidistant w.r.t multiple points.

With the proposal logic-regularized-loss method in Section 4, the proposed algebraic learning framework sheds new light on the area of neural-symbolic integration. More specifically, we proposed a method to integrate chain-like logic rules of relations into distributed representations. However, this only covers a small set of general logic, and it is, therefore, interesting to develop further methods to integrate other types of logic rules, including ones concerning entity attributes (also called arity-1 relations). Furthermore, the current work focuses merely on relation embeddings, which have an algebraic nature. The entity embedding, on the other hand, plays the role of "action space" of the relational

algebra and therefore has a geometric nature. Given an algebraic structure, the choice of its "action space" is far from being fully determined. There is hence a rich set of candidates for entity embedding design, which is worth to investigate in the future.

6. Conclusions

The mutual dependence of relations in a knowledge graph suggests the existence of an algebraic structure, which is introduced in this work as *Knowledgebra*. By analyzing a general KG based on five distinct properties, we determined that the semigroup is the most reasonable algebraic structure for general relation embeddings, where only totality and associativity are required. Our theoretical analysis was based on the work of NagE [27], and differed from it majorly by demonstrating that invertibility should be allowed to break. In Section 2.3, we provided proof based on contradictions with several examples among kinship relations. With the instantiation model, SemE proposed, we could discuss the invertibility issue from an alternative perspective.

Author Contributions: Conceptualization, T.Y., Y.W. and L.S.; methodology, T.Y., Y.W. and L.S.; software, T.Y., Y.W. and L.S.; validation, T.Y., Y.W. and L.S.; formal analysis, T.Y., Y.W. and L.S.; investigation, T.Y., Y.W. and L.S.; resources, Y.W and P.H.; writing—original draft preparation, T.Y., Y.W. and L.S.; writing—review and editing, T.Y., Y.W., L.S., J.E. and P.H.; supervision, J.E. and P.H.; project administration, J.E. and P.H.; funding acquisition, P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSF OAC-1920147 and NSF DMR-1933525 and the APC was also funded by NSF OAC-1920147 and NSF DMR-1933525.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

KG	Knowledge Graph
KGE	Knowledge Graph Embedding

References

1. Guo, Q.; Zhuang, F.; Qin, C.; Zhu, H.; Xie, X.; Xiong, H.; He, Q. A Survey on Knowledge Graph-Based Recommender Systems. *IEEE Trans. Knowl. Data Eng.* **2020**, *1*, 5555. [CrossRef]
2. Bordes, A.; Weston, J.; Usunier, N. Open question answering with weakly supervised embedding models. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, France, 15–19 September 2014; pp. 165–180.
3. Bordes, A.; Chopra, S.; Weston, J. Question Answering with Subgraph Embeddings. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 615–620. [CrossRef]
4. Huang, X.; Zhang, J.; Li, D.; Li, P. Knowledge graph embedding based question answering. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; pp. 105–113.
5. Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; Weld, D.S. Knowledge-based weak supervision for information extraction of overlapping relations. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 541–550.
6. Daiber, J.; Jakob, M.; Hokamp, C.; Mendes, P.N. Improving efficiency and accuracy in multilingual entity extraction. In Proceedings of the 9th International Conference on Semantic Systems, Graz, Austria, 4–6 September 2013; pp. 121–124.
7. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583–589. [CrossRef] [PubMed]
8. Thakur, N.; Han, C.Y. A Study of Fall Detection in Assisted Living: Identifying and Improving the Optimal Machine Learning Method. *J. Sens. Actuator Netw.* **2021**, *10*, 39. [CrossRef]
9. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.

10. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
11. Chami, I.; Wolf, A.; Juan, D.C.; Sala, F.; Ravi, S.; Ré, C. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6901–6914. [CrossRef]
12. Bordes, A.; Usunier, N.; Garcia-Durán, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13), Mountain View, CA, USA, 6–12 December 2020; Volume 2, pp. 2787–2795.
13. Fan, M.; Zhou, Q.; Chang, E.; Zheng, F. Transition-based knowledge graph embedding with relational mapping properties. In Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing, Phuket, Thailand, 12–14 December 2014; pp. 328–337.
14. Xiao, H.; Huang, M.; Zhu, X. From One Point to a Manifold: Knowledge Graph Embedding for Precise Link Prediction. In Proceedings of the IJCAI'16, New York, NY, USA, 9–15 July 2016; pp. 1315–1321.
15. Feng, J.; Huang, M.; Wang, M.; Zhou, M.; Hao, Y.; Zhu, X. Knowledge graph embedding by flexible translation. In Proceedings of the Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning, Cape Town, South Africa, 25–29 April 2016.
16. Xiao, H.; Huang, M.; Hao, Y.; Zhu, X. TransA: An adaptive approach for knowledge graph embedding. *arXiv* **2015**, arXiv:1509.05490.
17. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28.
18. 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.
19. 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; pp. 687–696.
20. Guo, S.; Wang, Q.; Wang, L.; Wang, B.; Guo, L. Jointly embedding knowledge graphs and logical rules. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 192–202.
21. Guo, S.; Wang, Q.; Wang, L.; Wang, B.; Guo, L. Knowledge graph embedding with iterative guidance from soft rules. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
22. Cheng, K.; Yang, Z.; Zhang, M.; Sun, Y. UniKER: A Unified Framework for Combining Embedding and Definite Horn Rule Reasoning for Knowledge Graph Inference. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 9753–9771.
23. Qu, M.; Tang, J. Probabilistic logic neural networks for reasoning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–14.
24. Harsha Vardhan, L.V.; Jia, G.; Kok, S. Probabilistic logic graph attention networks for reasoning. In Proceedings of the Companion Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 669–673.
25. Zhang, Y.; Chen, X.; Yang, Y.; Ramamurthy, A.; Li, B.; Qi, Y.; Song, L. Can graph neural networks help logic reasoning? *arXiv* **2019**, arXiv:1906.02111.
26. Xu, C.; Li, R. Relation Embedding with Dihedral Group in Knowledge Graph. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 263–272.
27. Yang, T.; Sha, L.; Hong, P. NagE: Non-Abelian Group Embedding for Knowledge Graphs. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Online, 19–23 October 2020. doi: 10.1145/3340531.3411875. [CrossRef]
28. Barr, M.; Wells, C. Toposes, Triples, and Theories. 1985. Available online: https://books.google.com.hk/books?id=q_-EAAAAIAAJ (accessed on 15 January 2022).
29. Wikipedia Contributors. Category (Mathematics)—Wikipedia. The Free Encyclopedia. 2021. Available online: https://en.wikipedia.org/w/index.php?title=Category_(mathematics)&oldid=1053436352 (accessed on 1 March 2022).
30. Choudhary, S.; Luthra, T.; Mittal, A.; Singh, R. A survey of knowledge graph embedding and their applications. *arXiv* **2021**, arXiv:2107.07842.
31. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Berg, R.v.d.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the European Semantic Web Conference, Monterey, CA, USA, 8–12 October 2018; pp. 593–607.
32. Balazevic, I.; Allen, C.; Hospedales, T. TuckER: Tensor Factorization for Knowledge Graph Completion. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5185–5194. [CrossRef]
33. 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.
34. Toutanova, K.; Chen, D. Observed versus latent features for knowledge base and text inference. In Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, Beijing, China, 26–31 July 2015; pp. 57–66.
35. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]

36. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250.
37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning (PMLR), New York, NY, USA, 20–22 June 2016; pp. 2071–2080.
39. 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.