HAL
open science

KEMA: Knowledge-Graph Embedding Using Modular Arithmetic

Hussein Baalbaki, Hussein Hazimeh, Hassan Harb, Rafael Angarita

To cite this version:

Hussein Baalbaki, Hussein Hazimeh, Hassan Harb, Rafael Angarita. KEMA: Knowledge-Graph Embedding Using Modular Arithmetic. The 34th International Conference on Software Engineering and Knowledge Engineering, Jul 2022, Pittsburgh, United States. hal-03657976

HAL Id: hal-03657976

https://hal.science/hal-03657976

Submitted on 16 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

KEMA: Knowledge-Graph Embedding Using Modular Arithmetic

Hussein Baalbaki
ISEP
Sorbonne University
Paris, France
hussein.baalbaki@ext.isep.fr

Hussein Hazimeh and Hassan Harb
Faculty of Sciences
Lebanese University
Beirut, Lebanon
hussein.hazimeh@ul.edu.lb, hassan.harb.1@ul.edu.lb

Rafael Angarita
ISEP
Sorbonne University
Paris, France
rafael.angarita@isep.fr

***Abstract*—Knowledge graph is a knowledge representation technique that helps representing entities and relations in a machine understandable way. This promising trend suffers from the problem of incompleteness that was best solved by link prediction. Indeed, link prediction is the most successful method for understanding the structure of the large knowledge graphs. Knowledge graph embedding KGE is one of the best link prediction methods. Its effectiveness is mainly affected by the accuracy of learning representations of entities and relations. In this paper, we propose a new knowledge graph embedding model called KEMA (Knowledge-graph Embedding using Modular Arithmetic). KEMA has the ability to represent simple and complex relations in an efficient way. Consequently, this allows our model to outperform the majority of the existing models. Mainly, KEMA depends on representing the relations in a knowledge graph by modular arithmetic operations applied between entities. Experimental results on multiple benchmark knowledge graphs verify the accurate representation, low complexity and scalability of KEMA.**

Keywords; Knowledge Graph Embedding, Knowledge Graphs, Link Prediction, Reasoning, Modular Arithmetic.

I. Introduction

Knowledge graph (KG) rises recently as one of the best ways for knowledge representation. We have seen the construction of many KGs of different sizes, domains, and coverage, like Freebase [1], Yago [2], and WordNet [3]. KG is a multi-relational graph built of nodes representing real world entities such as objects and events. These entities are connected by edges representing the relations and interactions between them. KG is represented by a set of triplets that shows the relations linking the entities. KG has proven to be effective in many real-world applications like recommender systems [4], natural language processing [5], and question-answering [6].

Although a KG may consist of a huge number of entities and relations, it is usually incomplete. It is impossible for a KG to cover every single entity or relation in whole world, no matter how huge this KG is. This is called *the completeness problem* of KG. This challenge represents one of the main issues facing KGs that researchers are working to solve. Link prediction emerges as efficient way to overcome the KG completeness problem. Subsequently, link prediction is used to predict not only the existence of a relation between two entities, but also the specific type of this relation. However, these predictions are infeasible using traditional methods. So the need for novel link prediction approaches like Knowledge graph embedding arises. Knowledge graph embedding (KGE) methods have proven to be very effective applied in link prediction. KGE embeds a KG into a continuous vector space while preserving certain information of the graph. Generally, KGE replaces any object (entity, relation, ..) with a vector of continuous numbers holding this object semantics. Mainly, KGE models differ in how these numeric vectors are used. and divided into three categories accordingly. The first one is the Translational that consider relations as a motion between entities. The second category is Tensor factorization that uses tensors for embedding vectors processing. The third category is the Neural network, in which the embedding vectors are fed to neural networks for training.

In this paper, we introduce a novel knowledge graph embedding model based on the modulus operation called as KEMA. Indeed, KEMA adopts a new way for dealing with embedding vectors away from translational, tensor factorization, and neural networks. Our model relies on the modular arithmetic mathematical operation. Since modular arithmetic is an equivalence relation, it helps handling different types of knowledge graph relations such as symmetry, inversion, and composition. Moreover, KEMA can deal with relations of complex mapping patterns like one-to-many, many-to-one, and many-to-many. To prove the effectiveness of our proposed model, we evaluate KEMA on a set of knowledge graph benchmark datasets including FB15k-237 [7] and WN18RR [8], and we compared the results to state-of-the-art approaches.

The rest of the paper is organized as follows. Section II gives an overview about different knowledge graph embedding models existing in the literature. Section III explains the modular arithmetic mathematical operation and its working way. Section IV presents KEMA, our proposed model. The obtained simulation results are exposed in Section VI. Finally, we conclude the paper and provide directions for future work in section VII.

II. Related Work

In the literature, we can distinguish between three types of knowledge graph embedding categories [9]: (1) Translation-based models; (2) Tensor factorization-based models; (3) Neural network-based models.

A. Translation-based models

The first translational-based model is called TransE [10]. Typically, TransE considers the relation between two entities as a translation operation in embedding space that starts by the head entity and leads to the tail. A score function f is used to compute the authenticity of the given triplet (h, r, t): $f_r(h, t) = |h+r-t|$. Although TransE shows a high efficiency when applied in large-scale knowledge graph embedding, it still struggle when dealing with complex relations such as $1 - N, N - 1$, and $N - N$. In order to overcome such challenge, the authors of [11] proposed an extension of TransE called TransH. Indeed, TransH assigns a hyper-plane to each relation, so that the heads and the tails of this relation are projected to. This model enables each entity to have different embedding representations depending on the relation involved in. In [12], TransR focuses on the entities carrying various semantic meanings. As a result, TransR expands the relation-specific hyper-planes concept of TransH to relation-specific spaces. Compared to transE and transH, transR makes a significant performance improvement but still has some gaps. One TransR weakness is sharing one projection matrix by both head and tail. The attributes of the same entity may differ according to its position in a relation (head / tail). To solve this problem, TransD [12] solves this problem by creates two different projection matrices for each single relation. The first matrix is used for head projection and the other is for the tail projection. Lastly, the authors of [13] proposed RotatE that represents the relation between two entities as rotation motion that starts from the head entity and ends by the tail.

B. Tensor factorization-based models

The idea behind tensor factorization-based models is based on representing all the triplets of a knowledge graph in a 3D binary tensor X. Two dimensions of X are of size n, the number of the entities in KG, while the third dimension is of size m, the number of relations in KG. S_r is $(n*n)$ slice of X that represents the relation r. The index $S_r[head][tail]$ of the slice S_r is filled with 1 if the relation between the head and the tail entities holds, otherwise it is filled by 0. The obtained tensor X is then broken down using factorization into a set of embedding matrices that are assigned to entities and relations. RESCAL [14] uses $Rank - d$ factorization to obtain three matrices:

$$X = ARA^T, \tag{1}$$

where A is a 2D matrix carrying the semantics of the KG entities, A^T is its transpose, and R is the 3D embedding matrix of the relations in KG. Accordingly, the scoring function of a given triplet (h, r, t) is calculated as follow:

$$f_r(h, r) = h^T m_r t, \tag{2}$$

where h^T is the transpose of h embedding vector, m_r is a 2D slice of R holding the relation r embedding matrix, and t is the tail embedding vector. To reduce the calculation complexity, a simplified version of RESCAL called DistMult is proposed in [15].In DistMult, any relation matrix m_r, obtained after factorization is considered to be diagonal with a score function:

$$F = h^T diag(m_r)t. \tag{3}$$

This step not only facilitates calculations, but it also shows improvement in terms of performance. Moreover, in order to capture the pairwise compositional representations of entities, HolE is proposed in [16]. Basically, HolE relies on a circular correlation with the following score function

$$f_r(h, t) = r^T(h \star t) \tag{4}$$

HolE reduces the composite representation complexity compared to tensor product. Lastly, ComplEx [17] was proposed as extension of DistMult, where it closes the gap resulting from the inability of DistMult to deal with asymmetric relations by integrating a complex space as follows:

$$fr(h, t) = Re(h_t * diag(r) - t) \tag{5}$$

where Re(·) denotes the real part of a complex value, and t represents the complex conjugate of t.

C. Neural network-based models

Neural networks are used for expressing complex nonlinear projections. They become recently a hot topic used to embed a knowledge graph into a continuous feature space. Particularly, Semantic Matching Energy SME [18] is a neural network-based model that calculates the energy of a given triplet (h, r, t) by applying two projection matrices, M_{left} is applied for the head h and the relation r embedding vectors, while M_{right} is applied for relation r and the tail t embedding vectors. Then, the results are given to a fully connected layer that returns the score of the semantic matching energy. ConvKB [19] is another model that uses convolutional neural network (CNN) to capture the latent semantic information in the triplets. In such model, the embedding vectors of the elements of a triplet are combined to form a matrix. Then, the matrix is fed to a convolution layer to produce multiple feature maps. Finally, these feature maps are concatenated and projected to a score that is used to estimate the authenticity of the triplet. Neural Association Model NAM [20] utilizes a deep neural network structure to represent a KG. After representing each element of a triplet by embedding vector, it concatenates the head entity vector and the relation vector to a single vector. The single vector is then fed to the next layer. Finally, NAM calculates the score by applying the output of the last hidden layer z^L with the tail embedding vector t:

$$f_r(h, t) = \sigma(z^L t) \tag{6}$$

where $\sigma(\cdot)$ is a sigmoid activation function.

III. Our Embedding Model: KEMA

Knowledge graph embedding is the process of representing the entities and relations of a given knowledge graph using numerical vectors. In this way, mathematical operations can be applied to these vectors in order to help studying and predicting the links connecting the entities. In this paper, we introduce

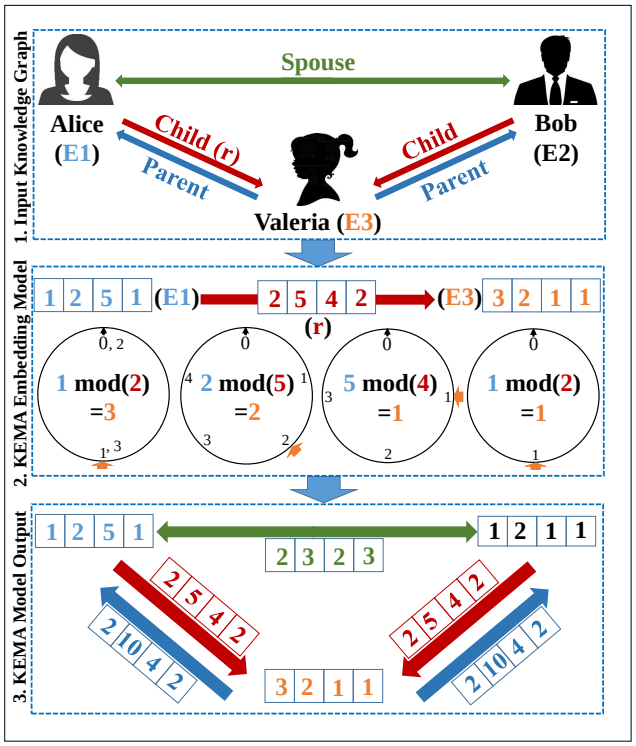

Fig. 1. KEMA architecture.

a new embedding KG approach called KEMA. Indeed, KEMA does not follow any of the classifications detailed in the related work section. It relies on one simple mathematical operation called modular arithmetic. The objective of KEMA is to predict the missing links connecting the entities of a given knowledge graph. As shown in Fig. 1, first, the input knowledge graph layer receives a knowledge graph as input. Then, KEMA embedding layer processes the knowledge graph components and embeds it to a low dimension continuous space. Finally, KEMA output layer returns a representative numerical vector for every entity and relation in the KG. These representative numerical vectors are then used for predicting links between KG entities.

A. Modular arithmetic

Modular arithmetic is a system of arithmetic that replaces all the numbers by their remainders of its division to a fixed number. Subsequently, the fixed number is an integer called "modulus". In modular arithmetic, every value of modulus m can be considered as a representative space $mod(m)$. In this space, we can represent every integer i by its remainder r resulting from its division by m, as shown in Equation (7). Since all the remainders of the division by m are smaller than m itself, then all the representative integers in $mod(m)$ space are smaller than m (Equation (8)).

$$i = m * x + r \iff r = i \ mod(m), \quad \forall \ i \in \ \mathbb{Z}, \tag{7}$$

$$r = i \ mod(m) \iff r \in]-m, m[\tag{8}$$

where $m, r, x \in \mathbb{Z}$

Indeed, an important example to illustrate the process of the modular arithmetic in this paper is 12-hour clock. In such example, the modulus value is 12 and the day is divided into two 12-hour periods. Whenever the hours count exceeds 12, it wraps around, and returns the remainder value.

Usually, modular arithmetic produces a set of integers having the same remainder when dividing by m. These integers are considered equal in the $mod(m)$ space, and are said to be congruent ($\equiv$), as shown in Equation (9).

$$\begin{aligned} x \ mod(m) &= y \\ z \ mod(m) &= y \\ x \ &\equiv \ z \end{aligned} \tag{9}$$

B. KEMA Embedding Model

The novel idea behind KEMA is to represent the relation between two entities through modular arithmetic operation. In other words, the tail embedding vector is considered to be the projection of the head embedding vector in the modular arithmetic space of the relation embedding vector, as shown in Equation (10).

$$t \ = \ h \ mod \ (r) \tag{10}$$

where h, t, and r represent the embedding vectors of the head, the tail and the relation respectively.

The second layer of our model is called KEMA embedding model (see Fig. 1), and it shows the way the embedding vectors are assigned for entities and relations of a given KG. KEMA starts by assigning random vectors for entities and relations. Then, it modifies these vectors in a way it satisfies the score function shown in Equation (10). First, every index $E_h[j]$ in the vector of the head entity E_h is subjected to modular arithmetic operation of modulus $r[j]$, the j-th index of relation r. Then, the vector of numbers obtained from this operation is assigned to the tail entity E_3 of the relation r.

C. Types of Relations

Despite the simplicity of the calculation process used in KEMA, it has proved to be highly effective and accurate compared to other models. This simplicity can also be seen in the low complexity of both training and prediction processes. As well as simple relations, KEMA can effectively handle complex relations of KG such as 1-N and N-N.

1) Simple Relations: Simple relation is that connecting no more than two entities, the head and the tail. According to the existing literature, three types of simple relation patterns are very important: symmetric, inverse, and composed patterns. All these patterns are covered by KEMA embedding model as follows:

- *Symmetric Relation*: This relation, switching between the head and the tail entities of a relation is possible. A relation r is said to be symmetric, if $\forall x, y \in E$, the set

of entities

$$(x, r, y) \implies (y, r, x) \tag{11}$$

- *Inverse Relation*: A relation r_2 is said to be inverse of relation r_1 whenever r_1 and r_2 have opposite directions connecting the same entities. A relation r_2 is the inverse of r_1, if $\forall x, y \in E$, the set of entities in KG

$$(x, r_1, y) \implies (y, r_2, x) \tag{12}$$

- *Composed Relation*: A relation r is said to be composed, if it can be broken down into two relations or more. A relation r is a composed relation, if $\exists\ r_1, r_2 \in R$, the set of relations in KG

$$(x, r, z) + (z, r, y) \implies (x, r, y) \tag{13}$$

Where $x, y, z \in E$, the set of entities.

For Modular Arithmetic

Let $a, b, n \in \mathbb{Z}$ such that:
$a \equiv b(mod\ n)$
$\implies a - b = kn$, for some $k \in \mathbb{Z}$
$\implies b - a = (-k)n$ and $-k \in \mathbb{Z}$
$\implies b \equiv a(mod\ n)$
Thus modular arithmetic is a symmetric relation.

Let $a, b, n \in \mathbb{Z}$ such that:
$a \equiv b(mod\ n)$
$\implies a - b = kn$, for some $k \in \mathbb{Z}$
$\implies b - a = (-n)k$ and $-n \in \mathbb{Z}$
$\implies b \equiv a(mod\ k)$
mod(n) is inverse to mod(k).
Thus modular arithmetic is an inverse relation.

Let $a, b, n, c \in \mathbb{Z}$, such that:
$a \equiv b(mod\ n)$ and $b \equiv c(mod\ n)$.
then $a = b + kn, k \in \mathbb{Z}$ and $b = c + hn, h \in \mathbb{Z}$.
$a = b + kn$
$\implies a = (c + hn) + kn$
$\implies a = c + (hn + kn)$
$\implies a = c + (h + k)n,\ h + k \in Z$.
Hence $a \equiv c(mod\ n)$.
Thus modular arithmetic is a composed relation.

---

end

2) KEMA Complex Relations: The majority of the models proposed in the literature can deal with simple relations between KG entities, i.e 1-to-1 relations. However, relying on simple relationships to build knowledge is impractical. Contrarily, KEMA has the ability to handle both simple and complex relationships.

Giving a 1-to-N relationship r, the set of tails $T = \{t_1, t_2, ..t_N\}$ can share a unique head whenever all these tails are congruent in $mod(r)$ space, Equation (14). Similarly, the set of heads $H = \{h_1, h_2, ..h_N\}$ can share the same tail whenever all these heads are congruent in $mod(r)$ space, Equation (15). Moreover, KEMA allows the representation of N-to-N complex relationships, in which one relation can have several heads and tails at once, by combining both equations (14) and (15)

TABLE I
EMBEDDING VECTORS OF ENTITIES AND RELATIONS

Entity	Embedding vector	Relation	Embedding vector
Bob	[1,2,1,1]	$Spouse$	[2,3,2,3]
$Alice$	[1,2,5,1]	$Child$	[2,5,4,2]
$Valeria$	[3,2,1,1]	$Parent$	[2,4,10,2]

$$t_1 \equiv t_2\ ..\ \equiv t_N \iff h = t\ mod\ (r),\ \forall\ t \in T, \tag{14}$$
$$h_1 \equiv h_2\ ..\ \equiv h_N \iff t = h\ mod\ (r),\ \forall\ h \in H, \tag{15}$$

For Modular Arithmetic

Let $a, b, n, c \in \mathbb{Z}$ such that:
$a \equiv b(mod\ n)$

then $a = b + kn, \quad k \in \mathbb{Z}$
$\implies a = b + (k - c + c)n$
$\implies a = b + cn + (k - c)n, \quad k - c \in Z$.
Hence $a \equiv b + cn(mod\ n)$ and $a \equiv b(mod\ n)$

Thus modular arithmetic holds for 1-N relations.

Given the 1-N relation:

$$a \equiv b + cn(mod\ n) \text{ and } a \equiv b(mod\ n)$$

Since modular arithmetic is symmetric relation, then:

$$b + cn \equiv a(mod\ n) \text{ and } b \equiv a(mod\ n)$$

Thus modular arithmetic can represent N-1 relations.

By combining the 1-N and N-1 modular arithmetic relations, we conclude its ability to represent N-N relation, and thus modulus can represent all the complex relation patterns.

D. Analytical example

In this section, we illustrate an example to show the effectiveness of KEMA in terms of representing simple and complex relations. According to the input knowledge graph layer shown in figure 1, the sub graph shows the relations between three entities. Table I shows the embedding vectors that KEMA assigned to every entity and relation.

The relation "$Spouse$" is an example of the symmetric relation. In Fig. 1, the output layer of KEMA shows that this relation holds in both directions. Moreover, in Table II, the first row shows that the tail of the relation "Spouse" with head entity "Bob" is "Alice". On the other hand, the second row represents the opposite direction, where the tail of the relation "Spouse" with head entity "Alice" is "Bob".

TABLE II
SYMMETRIC RELATION EXAMPLE

Head	Relation	Tail (result)
[1,2,1,1]	[2,3,2,3]	[1,2,5,1]
[1,2,5,1]	[2,3,2,3]	[1,2,1,1]

TABLE III
INVERSE RELATIONS EXAMPLE

Head	Relation	Tail (result)
[1,2,5,1]	[2,4,10,2]	[3,2,1,1]
[3,2,1,1]	[2,4,5,2]	[1,2,5,1]

In Fig. 1, the relations "Child" and "Parent" show the inversion pattern of simple relations. In the first row of the Table III, "Alice" is the head of the relation "Child" while "Valeria" is its tail. In the opposite direction, the second row shows the relation "Parent", where "Valeria" is the head and "Alice" is the tail. Then the relations "Parent" and "Child" are said to be inverse.

Furthermore, Fig. 1 shows that the relation "Spouse" is a composed relation. Fig. 1 shows that the relation "Child" of head "Alice" and tail "Valeria", followed by the relation "Parent" of head "Valeria" and tail "Bob", can be replaced by the relation "Spouse" having the same head as "Child", and the same tail as "Parent".

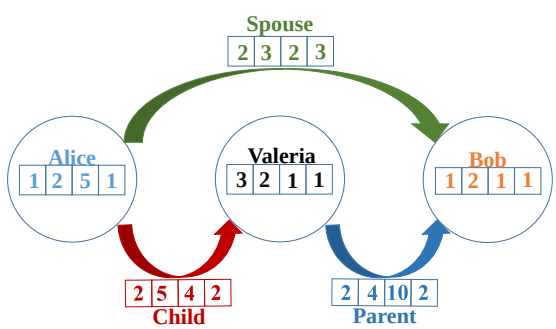

Fig. 2. Composed relation example.

Indeed, the strength of KEMA in terms of representing complex relations is shown through the relations "Parent" and "Child" in Fig. 1. Since "Parent" has one head which is "Valeria", and two tails which are "Bob" and "Alice", then it is a complex relation of $1-N$ mapping pattern. Whereas "Child" relation has two heads "Bob" and "Alice" connected to one tail "Valeria", representing N-1 pattern.

IV. SIMULATION RESULTS

In this section, we will show the experiment setting to implement our model followed by the discussion of the obtained results.

A. Experimental setting

To evaluate our model, we implemented KEMA using Ampligraph python library. Then, we compare it to the state-of-the-art models on two commonly used benchmark datasets: WN18RR, and FB15K-237.

- WN18RR is a subset of WordNet, a KG that clusters words into synonym groups and features lexical relationships between words. It consists of 40,943 entities, 11 relation types, and 93003 triples. WN18RR contains symmetry, antisymmetry and composition relation patterns. The main pattern is the symmetry since almost each word has a symmetric relation in WN18RR, e.g., $also-see$ and $similar-to$ [13].
- FB15k-237 is a subset of Freebase, a large knowledge graph that stores general knowledge facts. It consists of 14951 entities, 237 relation types, and 310116 triples. The main patterns of the relation in FB15k are symmetry, antisymmetry and composition [13].

To train and evaluate our model, we need to use negative triples, which are not available in both WN18RR and FB15K-237. As a result, we used to corrupt the positive triples of the datasets to generate negative samples for each positive. This is what is called the local closed world assumption. That is, for a triple, we randomly replace the entity in the subject or the object position by another, but not both at once.

We applied three tests to evaluate the performance of link prediction of KEMA. These tests rely on ranking each positive test triple against all its generated negatives according to its score. The first test is Mean Rank (MR) which is calculated as follow:

$$mean(rank_t)\ \forall t \in T \tag{16}$$

with T is the set of positive test triples, and $rank_t$ is the rank of triple t against its negatives.

The second evaluation test is Mean Reciprocal Rank (MRR). It is similar to MR, but it uses the reciprocal rank of a triple instead of its rank, what make it less sensitive to outliers [16]:

$$mean(1/rank_t)\ \forall t \in T \tag{17}$$

The last evaluation test is Hits@N, which counts the test triples having a rank less than or equal to N.

$$\sum t_N,\ where\ t_N \in T, rank_{t_N} >= N \tag{18}$$

B. Main Results

In our simulation, we compared KEMA to several state-of-the-art models including TransE [10], DistMult [15], ComplEx [17], ConvE [8], and RotatE [13]. We show the efficiency of our proposed model inferring the relation patterns for the task of predicting missing links. Tables IV shows the results of the evaluation tests of our model and the state-of-the-art models based on WN18RR and FB15K-237 datasets respectively.

FB15K-237 dataset contains symmetry, anti-symmetry and composition relation patterns. The main pattern in this dataset is the composition [13]. The domination of the composition pattern can be inferred from the results shown in Table IV. So that Table IV shows that the model TransE, representing composition and anti-symmetry patterns, outperforms ComplEx model representing symmetry and anti-symmetry patterns.

TABLE IV
RESULTS OF MODELS EVALUATION ON WN18RR AND FB15K-237 DATASETS

	WN18RR					FB15K-237				
Model	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
TransE	3384	0.226	-	-	0.501	357	0.294	-	-	0.465
DisMult	5110	0.43	0.39	0.44	0.49	254	0.241	0.155	0.263	0.419
ComplEx	5261	0.44	0.41	0.46	0.51	339	0.247	0.158	0.275	0.428
ConvE	4187	0.43	0.40	0.44	0.52	244	0.325	0.237	0.356	0.501
RotatE	3340	0.476	0.428	0.492	0.571	177	0.338	0.241	0.375	0.533
KEMA	**1898**	**0.477**	**0.442**	**0.486**	**0.543**	**244**	**0.311**	**0.223**	**0.342**	**0.486**

Furthermore, Table IV shows the superiority of KEMA over TransE, DistMult, and ComplEx in all the tests, due to its ability to perfectly represent composition and symmetry patterns. On the other hand, ConvE and RotatE surpass KEMA due to its ability to represent all the relation patterns of FB15K-237, which is composition, symmetry, and anti-symmetry.

Table IV shows the results of the evaluation tests of KEMA and the state-of-the-art models on WN18RR dataset. Similar to FB15K-237 dataset, WN18RR contains composition, symmetry and anti-symmetry relation patterns. Symmetry is the dominating pattern in this dataset. This conclusion can be easily inferred from the Table IV. So that the model DistMult, representing only symmetric relations, performs better than TransE representing both anti-symmetry and composition patterns. On WN18RR dataset, our model outperforms all the state-of-the-art models. Although RotatE and ConvE both represent all the relation patterns contained in WN18RR, KEMA surpasses both models that confirms its high performance.

V. CONCLUSION AND FUTURE WORK

Link prediction is among the most prominent methods that solve the problem of incompleteness of Knowledge graph. It is used to predict the existence and the type of a relation connecting two entities. The more the knowledge graph is well represented, the more the predictions are accurate. In this paper, We proposed a novel knowledge graph embedding model (KEMA) that relies on modular arithmetic operation in representing relations between entities. KEMA applies modular arithmetic to the head entity with modulus equal to the relation vector. The main strength of our model lies in its ability to represent complex relations like one-to-many, in addition to representing symmetry, inverse, and composed simple relations. The results of our experiments show that KEMA outperforms the majority of the existing models in representation accuracy while preserving low level of complexity.

In the future work, we plan to build a complete KEMA framework that contains beside the proposed model a loss function and a suitable negative sampling method.

REFERENCES

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

[2] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.

[3] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[4] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 353–362.

[5] B. Yang and T. Mitchell, "Leveraging knowledge bases in lstms for improving machine reading," *arXiv preprint arXiv:1902.09091*, 2019.

[6] Y. Hao, Y. Zhang, K. Liu, S. He, Z. Liu, H. Wu, and J. Zhao, "An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 221–231.

[7] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, 2015, pp. 57–66.

[8] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[9] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, p. 750, 2020.

[10] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, pp. 1–9, 2013.

[11] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014.

[12] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "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 (Volume 1: Long Papers)*, 2015, pp. 687–696.

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

[14] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Icml*, 2011.

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

[16] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

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

[18] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, 2014.

[19] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," *arXiv preprint arXiv:1712.02121*, 2017.

[20] Q. Liu, H. Jiang, A. Evdokimov, Z.-H. Ling, X. Zhu, S. Wei, and Y. Hu, "Probabilistic reasoning via deep learning: Neural association models," *arXiv preprint arXiv:1603.07704*, 2016.