

NeuInfer: Knowledge Inference on N-ary Facts

Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng

CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences; School of Computer and Control Engineering, University of Chinese Academy of Sciences

{guansaiping, jinxiaolong, guojiafeng, wangyuanzhuo, cxq}@ict.ac.cn

Abstract

Knowledge inference on knowledge graph has attracted extensive attention, which aims to find out connotative valid facts in knowledge graph and is very helpful for improving the performance of many downstream applications. However, researchers have mainly poured attention to knowledge inference on binary facts. The studies on n-ary facts are relatively scarcer, although they are also ubiquitous in the real world. Therefore, this paper addresses knowledge inference on n-ary facts. We represent each n-ary fact as a primary triple coupled with a set of its auxiliary descriptive attribute-value pair(s). We further propose a neural network model, NeuInfer, for knowledge inference on n-ary facts. Besides handling the common task to infer an unknown element in a whole fact, NeuInfer can cope with a new type of task, flexible knowledge inference. It aims to infer an unknown element in a partial fact consisting of the primary triple coupled with any number of its auxiliary description(s). Experimental results demonstrate the remarkable superiority of NeuInfer.

1 Introduction

With the introduction of connotative valid facts, knowledge inference on knowledge graph improves the performance of many downstream applications, such as vertical search and question answering (Dong et al., 2015; Lukovnikov et al., 2017). Existing studies (Nickel et al., 2016; Wang et al., 2017) mainly focus on knowledge inference on binary facts with two entities connected with a certain binary relation, represented as triples, (head entity, relation, tail entity). They attempt to infer the unknown head/tail entity or the unknown relation of a given binary fact. However, n-ary facts involving more than two entities are also ubiquitous. For example, in Freebase, more than 1/3 entities participate in n-ary facts (Wen et al., 2016). The

fact that *John Bardeen* received *Nobel Prize in Physics in 1956* together with *Walter Houser Brattain* and *William Shockley*¹ is a typical 5-ary fact. So far, only a few studies (Wen et al., 2016; Zhang et al., 2018; Guan et al., 2019) have tried to address knowledge inference on n-ary facts.

In existing studies for knowledge inference on n-ary facts, each n-ary fact is represented as a group of peer attributes and attribute values. In practice, for each n-ary fact, there is usually a primary triple (the main focus of the n-ary fact), and other attributes along with the corresponding attribute values are its auxiliary descriptions. Take the above 5-ary fact for example, the primary triple is (*John Bardeen*, *award-received*, *Nobel Prize in Physics*), and other attribute-value pairs including *point-in-time : 1956*, *together-with : Walter Houser Brattain* and *together-with : William Shockley* are its auxiliary descriptions. Actually, in YAGO (Suchanek et al., 2007) and Wikidata (Vrandečić and Krötzsch, 2014), a primary triple is identified for each n-ary fact.

The above 5-ary fact is a relatively complete example. In the real-world knowledge bases, many n-ary facts appear as only partial ones, each consisting of a primary triple and a subset of its auxiliary description(s), due to incomplete knowledge acquisition. For example, (*John Bardeen*, *award-received*, *Nobel Prize in Physics*) with *point-in-time : 1956* and it with {*together-with : Walter Houser Brattain*, *together-with : William Shockley*} are two typical partial facts corresponding to the above 5-ary fact. For differentiation, we call those relatively complete facts as whole ones. We noticed that existing studies on n-ary facts infer an unknown element in a well-defined whole fact and have not paid attention to knowledge inference on partial facts. Later on, we

¹<https://www.wikidata.org/wiki/Q949>

refer the former as simple knowledge inference, while the latter as flexible knowledge inference.

With these considerations in mind, in this paper, by discriminating the information in the same n-ary fact, we propose a neural network model, called NeuInfer, to conduct both simple and flexible knowledge inference on n-ary facts. Our specific contributions are summarized as:

- We treat the information in the same n-ary fact discriminatingly and represent each n-ary fact as a primary triple coupled with a set of its auxiliary descriptive attribute-value pair(s).
- We propose a neural network model, NeuInfer, for knowledge inference on n-ary facts. NeuInfer can particularly handle the new type of task, flexible knowledge inference, which infers an unknown element in a partial fact consisting of a primary triple and any number of its auxiliary description(s).
- Experimental results validate the significant effectiveness and superiority of NeuInfer.

2 Related Works

2.1 Knowledge Inference on Binary Facts

They can be divided into tensor/matrix based methods, translation based methods, and neural network based ones.

The quintessential one of tensor/matrix based methods is RESCAL (Nickel et al., 2011). It relates a knowledge graph to a three-way tensor of head entities, relations, and tail entities. The learned embeddings of entities and relations via minimizing the reconstruction error of the tensor are used to reconstruct the tensor. And binary facts corresponding to entries of large values are treated as valid. Similarly, ComplEx (Trouillon et al., 2016) relates each relation to a matrix of head and tail entities, which is decomposed and learned like RESCAL. To improve the embeddings and thus the performance of inference, researchers further introduce the constraints of entities and relations (Ding et al., 2018; Jain et al., 2018).

Translation based methods date back to TransE (Bordes et al., 2013). It views each valid binary fact as the translation from the head entity to the tail entity via their relation. Thus, the score function indicating the validity of the fact is defined based on the similarity between the translation result and the tail entity. Then, a flurry of methods

spring up (Wang et al., 2014; Lin et al., 2015b; Ji et al., 2015; Guo et al., 2015; Lin et al., 2015a; Xiao et al., 2016; Jia et al., 2016; Tay et al., 2017; Ebisu and Ichise, 2018; Chen et al., 2019). They modify the above translation assumption or introduce additional information and constraints. Among them, TransH (Wang et al., 2014) translates on relation-specific hyperplanes. Entities are projected into the hyperplanes of relations before translating.

Neural network based methods model the validity of binary facts or the inference processes. For example, ConvKB (Nguyen et al., 2018) treats each binary fact as a three-column matrix. This matrix is fed into a convolution layer, followed by a concatenation layer and a fully-connected layer to generate a validity score. Nathani et al. (2019) further proposes a generalized graph attention model as the encoder to capture neighborhood features and applies ConvKB as the decoder. ConvE (Dettmers et al., 2018) models entity inference process via 2D convolution over the reshaped then concatenated embedding of the known entity and relation. ConvR (Jiang et al., 2019) further adaptively constructs convolution filters from relation embedding and applies these filters across entity embedding to generate convolutional features. SENN (Guan et al., 2018) models the inference processes of head entities, tail entities, and relations via fully-connected neural networks, and integrates them into a unified framework.

2.2 Knowledge Inference on N-ary Facts

As aforesaid, only a few studies handle this type of knowledge inference. The m-TransH method (Wen et al., 2016) defines n-ary relations as the mappings from the attribute sequences to the attribute values. Each n-ary fact is an instance of the corresponding n-ary relation. Then, m-TransH generalizes TransH (Wang et al., 2014) on binary facts to n-ary facts via attaching each n-ary relation with a hyperplane. RAE (Zhang et al., 2018) further introduces the likelihood that two attribute values co-participate in a common n-ary fact, and adds the corresponding relatedness loss multiplied by a weight factor to the embedding loss of m-TransH. Specifically, RAE applies a fully-connected neural network to model the above likelihood. Differently, NaLP (Guan et al., 2019) represents each n-ary fact as a set of attribute-value pairs directly. Then, convolution is adopted to get the embeddings of the attribute-value pairs, and a fully-connected neural

network is applied to evaluate their relatedness and finally to obtain the validity score of the input n-ary fact.

In these methods, the information in the same n-ary fact is equal-status. Actually, in each n-ary fact, a primary triple can usually be identified with other information as its auxiliary description(s), as exemplified in Section 1. Moreover, these methods are deliberately designed only for the inference on whole facts. They have not tackled any distinct inference task. In practice, the newly proposed flexible knowledge inference is also prevalent.

3 Problem Statement

3.1 The Representation of N-ary Facts

Different from the studies that define n-ary relations first and then represent n-ary facts (Wen et al., 2016; Zhang et al., 2018), we represent each n-ary fact as a primary triple (head entity, relation, tail entity) coupled with a set of its auxiliary description(s) directly. Formally, given an n-ary fact Fct with the primary triple (h, r, t) , m attributes and attribute values, its representation is:

$$((h, r, t), \{ \begin{array}{l} \text{--- } a_1 : v_1, \\ \text{--- } a_2 : v_2, \\ \text{--- } \dots, \\ \text{--- } a_m : v_m \} \end{array}),$$

where each $a_i : v_i$ ($i = 1, 2, \dots, m$) is an attribute-value pair, also called an auxiliary description to the primary triple. An element of Fct refers to $h/r/t/a_i/v_i$; $A_{Fct} = \{a_1, a_2, \dots, a_m\}$ is Fct 's attribute set and a_i may be the same to a_j ($i, j = 1, 2, \dots, m, i \neq j$); $V_{Fct} = \{v_1, v_2, \dots, v_m\}$ is Fct 's attribute value set.

For example, the representation of the 5-ary fact, mentioned in Section 1, is:

$$((John\ Bardeen, award-received, Nobel\ Prize\ in\ Physics), \{ \begin{array}{l} \text{--- } point-in-time : 1956, \\ \text{--- } together-with : Walter\ Houser\ Brattain, \\ \text{--- } together-with : William\ Shockley \} \end{array}).$$

Note that, in the real world, there is a type of complicated cases, say, where more than two entities participate in the same n-ary fact with the same primary attribute. We follow Wikidata (Vrandečić and Krötzsch, 2014) to view the cases from different aspects of different entities. Take the case that *John Bardeen*, *Walter Houser Brattain*,

and *William Shockley* received *Nobel Prize in Physics* in 1956 for example, besides the above 5-ary fact from the view of *John Bardeen*, we get other two 5-ary facts from the views of *Walter Houser Brattain*² and *William Shockley*³, respectively:

$$((Walter\ Houser\ Brattain, award-received, Nobel\ Prize\ in\ Physics), \{ \begin{array}{l} \text{--- } point-in-time : 1956, \\ \text{--- } together-with : John\ Bardeen, \\ \text{--- } together-with : William\ Shockley \} \end{array}).$$

$$((William\ Shockley, award-received, Nobel\ Prize\ in\ Physics), \{ \begin{array}{l} \text{--- } point-in-time : 1956, \\ \text{--- } together-with : Walter\ Houser\ Brattain, \\ \text{--- } together-with : John\ Bardeen \} \end{array}).$$

3.2 Task Statement

In this paper, we handle both the common simple knowledge inference and the newly proposed flexible knowledge inference. Before giving their definitions under our representation form of n-ary facts, let us define whole fact and partial fact first.

Definition 1 (Whole fact and partial fact). *For the fact Fct , assume its set of auxiliary description(s) as $S_d = \{a_i : v_i | i = 1, 2, \dots, m\}$. Then a partial fact of Fct is: $Fct' = ((h, r, t), S'_d)$, where $S'_d \subset S_d$, i.e., S'_d is a subset of S_d . And we call Fct the whole fact to differentiate it from Fct' .*

Notably, whole fact and partial fact are relative concepts, and a whole fact is a relatively complete fact compared to its partial fact. In this paper, partial facts are introduced to imitate a typical open-world setting where different facts of the same type may have different numbers of attribute-value pair(s).

Definition 2 (Simple knowledge inference). *It aims to infer an unknown element in a whole fact.*

Definition 3 (Flexible knowledge inference). *It aims to infer an unknown element in a partial fact.*

4 The NeuInfer Method

4.1 The Framework of NeuInfer

To conduct knowledge inference on n-ary facts, NeuInfer first models the validity of the n-ary facts and then casts inference as a classification task.

²<https://www.wikidata.org/wiki/Q184577>

³<https://www.wikidata.org/wiki/Q163415>

4.1.1 The Motivation of NeuInfer

How to estimate whether an n-ary fact is valid or not? Let us look into two typical examples of invalid n-ary facts:

$((John\ Bardeen, award-received, Turing\ Award), \{$
 $\quad \text{--- } point-in-time : 1956,$
 $\quad \text{--- } together-with : Walter\ Houser\ Brattain,$
 $\quad \text{--- } together-with : William\ Shockley\})$.

$((John\ Bardeen, award-received, Nobel\ Prize$
 $\quad \text{in\ Physics}), \{$
 $\quad \text{--- } point-in-time : 1956,$
 $\quad \text{--- } together-with : Walter\ Houser\ Brattain,$
 $\quad \text{--- } place-of-marriage : New\ York\ City\})$.

In the above first n-ary fact, the primary triple is invalid. In the second one, some auxiliary description is incompatible with the primary triple.

Therefore, we believe that a valid n-ary fact has two prerequisites. On the one hand, its primary triple should be valid. If the primary triple is invalid, attaching any number of attribute-value pairs to it does not make the resulting n-ary fact valid; on the other hand, since each auxiliary description presents a qualifier to the primary triple, it should be compatible with the primary triple. Even if the primary triple is basically valid, any incompatible attribute-value pair makes the n-ary fact invalid. Therefore, NeuInfer is designed to characterize these two aspects and thus consists of two components corresponding to the validity evaluation of the primary triple and the compatibility evaluation of the n-ary fact, respectively.

4.1.2 The Framework of NeuInfer

The framework of NeuInfer is illustrated in Figure 1, with the 5-ary fact presented in Section 1 as an example.

For an n-ary fact Fct , we look up the embeddings of its relation r and the attributes in A_{Fct} from the embedding matrix $\mathbf{M}_R \in \mathbb{R}^{|R| \times k}$ of relations and attributes, where R is the set of all the relations and attributes, and k is the dimension of the latent vector space. The embeddings of h , t , and the attribute values in V_{Fct} are looked up from the embedding matrix $\mathbf{M}_E \in \mathbb{R}^{|E| \times k}$ of entities and attribute values, where E is the set of all the entities and attribute values. In what follows, the embeddings are denoted with the same letters but in boldface by convention. As presented in Figure 1, these embeddings are fed into the validity evaluation component (the upper part of Figure 1) and the

compatibility evaluation component (the bottom part of Figure 1) to compute the validity score of (h, r, t) and the compatibility score of Fct , respectively. These two scores are used to generate the final score of Fct by weighted sum \oplus and further compute the loss. Note that, following RAE (Zhang et al., 2018) and NaLP (Guan et al., 2019), we only apply fully-connected neural networks in NeuInfer.

4.2 Validity Evaluation

This component estimates the validity of (h, r, t) , including the acquisition of its interaction vector and the assessment of its validity, corresponding to “hrt-FCNs” and “FCN₁” in Figure 1, respectively.

Detailedly, the embeddings of h , r , and t are concatenated and fed into a fully-connected neural network. After layer-by-layer learning, the last layer outputs the interaction vector \mathbf{o}_{hrt} of (h, r, t) :

$$\mathbf{o}_{hrt} = f(f(\dots f(f([\mathbf{h}; \mathbf{r}; \mathbf{t}] \mathbf{W}_{1,1} + \mathbf{b}_{1,1}) \cdot \mathbf{W}_{1,2} + \mathbf{b}_{1,2}) \dots) \mathbf{W}_{1,n_1} + \mathbf{b}_{1,n_1}), \quad (1)$$

where $f(\cdot)$ is the ReLU function; n_1 is the number of the neural network layers; $\{\mathbf{W}_{1,1}, \mathbf{W}_{1,2}, \dots, \mathbf{W}_{1,n_1}\}$ and $\{\mathbf{b}_{1,1}, \mathbf{b}_{1,2}, \dots, \mathbf{b}_{1,n_1}\}$ are their weight matrices and bias vectors, respectively.

With \mathbf{o}_{hrt} as the input, the validity score val_{hrt} of (h, r, t) is computed via a fully-connected layer and then the sigmoid operation:

$$val_{hrt} = \sigma(\mathbf{o}_{hrt} \mathbf{W}_{val} + b_{val}), \quad (2)$$

where \mathbf{W}_{val} and b_{val} are the weight matrix and bias variable, respectively; $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, which constrains $val_{hrt} \in (0, 1)$.

For simplicity, the number of hidden nodes in each fully-connected layer of “hrt-FCNs” and “FCN₁” gradually reduces with the same difference between layers.

4.3 Compatibility Evaluation

This component estimates the compatibility of Fct . It contains three sub-processes, i.e., the capture of the interaction vector between (h, r, t) and each auxiliary description $a_i : v_i$ ($i = 1, 2, \dots, m$), the acquisition of the overall interaction vector, and the assessment of the compatibility of Fct , corresponding to “hrtav-FCNs”, “min” and “FCN₂” in Figure 1, respectively.

Similar to “hrt-FCNs”, we obtain the interaction vector $\mathbf{o}_{hrt a_i v_i}$ of (h, r, t) and $a_i : v_i$:

$$\mathbf{o}_{hrt a_i v_i} = f(f(\dots f(f([\mathbf{h}; \mathbf{r}; \mathbf{t}; \mathbf{a}_i; \mathbf{v}_i] \mathbf{W}_{2,1} + \mathbf{b}_{2,1}) \cdot \mathbf{W}_{2,2} + \mathbf{b}_{2,2}) \dots) \mathbf{W}_{2,n_2} + \mathbf{b}_{2,n_2}), \quad (3)$$

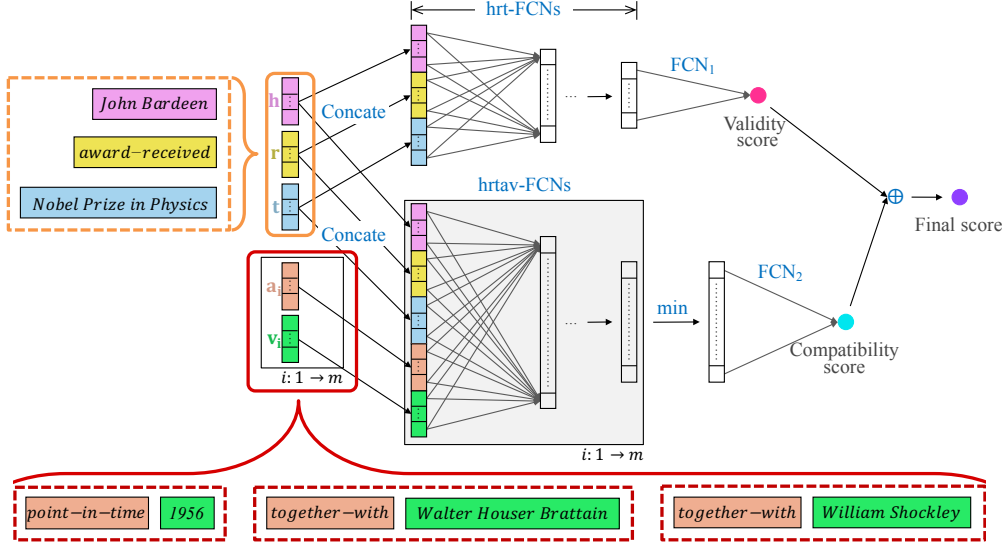


Figure 1: The framework of the proposed NeuInfer method.

where n_2 is the number of the neural network layers; $\{\mathbf{W}_{2,1}, \mathbf{W}_{2,2}, \dots, \mathbf{W}_{2,n_2}\}$ and $\{\mathbf{b}_{2,1}, \mathbf{b}_{2,2}, \dots, \mathbf{b}_{2,n_2}\}$ are their weight matrices and bias vectors, respectively. The number of hidden nodes in each fully-connected layer also gradually reduces with the same difference between layers. And the dimension of the resulting $\mathbf{o}_{\text{hrt}a_i v_i}$ is d . All the auxiliary descriptions share the same parameters in this sub-process.

The overall interaction vector $\mathbf{o}_{\text{hrtav}}$ of Fct is generated based on $\mathbf{o}_{\text{hrt}a_i v_i}$. Before introducing this sub-process, let us see the principle behind first.

Straightforwardly, if Fct is valid, (h, r, t) should be compatible with any of its auxiliary description. Then, the values of their interaction vector, measuring the compatibility in many different views, are all encouraged to be large. Therefore, for each dimension, the minimum over it of all the interaction vectors is not allowed to be too small.

Thus, the overall interaction vector $\mathbf{o}_{\text{hrtav}}$ of (h, r, t) and its auxiliary description(s) is:

$$\mathbf{o}_{\text{hrtav}} = \min_{i=1}^m (\mathbf{o}_{\text{hrt}a_i v_i}), \quad (4)$$

where $\min(\cdot)$ is the element-wise minimizing function.

Then, similar to “FCN₁”, we obtain the compatibility score $comp_{Fct}$ of Fct :

$$comp_{Fct} = \sigma(\mathbf{o}_{\text{hrtav}} \mathbf{W}_{\text{comp}} + b_{\text{comp}}), \quad (5)$$

where \mathbf{W}_{comp} of dimension $d \times 1$ and b_{comp} are the weight matrix and bias variable, respectively.

4.4 Final Score and Loss Function

The final score s_{Fct} of Fct is the weighted sum \oplus of the above validity score and compatibility score:

$$\begin{aligned} s_{Fct} &= val_{\text{hrt}} \oplus comp_{Fct} \\ &= w \cdot val_{\text{hrt}} + (1 - w) \cdot comp_{Fct}, \end{aligned} \quad (6)$$

where $w \in (0, 1)$ is the weight factor.

If the arity of Fct is 2, the final score is equal to the validity score of the primary triple (h, r, t) . Then, Equation (6) is reduced to:

$$s_{Fct} = val_{\text{hrt}}. \quad (7)$$

Currently, we obtain the final score s_{Fct} of Fct . In addition, Fct has its target score l_{Fct} . By comparing s_{Fct} with l_{Fct} , we get the binary cross-entropy loss:

$$\mathcal{L}_{Fct} = -l_{Fct} \log s_{Fct} - (1 - l_{Fct}) \log(1 - s_{Fct}), \quad (8)$$

where $l_{Fct} = 1$, if $Fct \in T$, otherwise $Fct \in T^-$, $l_{Fct} = 0$. Here, T is the training set and T^- is the set of negative samples constructed by corrupting the n-ary facts in T . Specifically, for each n-ary fact in T , we randomly replace one of its elements with a random element in E/R to generate one negative sample not contained in T .

We then optimize NeuInfer via backpropagation, and Adam (Kingma and Ba, 2015) with learning rate λ is used as the optimizer.

5 Experiments

5.1 Datasets and Metrics

We conduct experiments on two n-ary datasets. The first one is JF17K (Wen et al., 2016; Zhang

et al., 2018), derived from Freebase (Bollacker et al., 2008). In JF17K, an n-ary relation of a certain type is defined by a fixed number of ordered attributes. Then, any n-ary fact of this relation is denoted as an ordered sequence of attribute values corresponding to the attributes. For example, for all n-ary facts of the n-ary relation *olympics.olympic_medal_honor*, they all have four attribute values (e.g., 2008 Summer Olympics, United States, Natalie Coughlin, and Swimming at the 2008 Summer Olympics – Women’s 4 × 100 metre freestyle relay), corresponding to the four ordered attributes of this n-ary relation. The second one is WikiPeople (Guan et al., 2019), derived from Wikidata (Vrandečić and Krötzsch, 2014). Its n-ary facts are more diverse than JF17K’s. For example, for all n-ary facts that narrate *award-received*, some have the attribute *together-with*, while some others do not. Thus, WikiPeople is more difficult.

To run NeuInfer on JF17K and WikiPeople, we transform the representation of their n-ary facts. For JF17K, we need to convert each attribute value sequence of a specific n-ary relation to a primary triple coupled with a set of its auxiliary description(s). The core of this process is to determine the primary triple, formed by merging the two primary attributes of the n-ary relation and the corresponding attribute values. The two primary attributes are selected based on RAE (Zhang et al., 2018). For each attribute of the n-ary relation, we count the number of its distinct attribute values from all the n-ary facts of this relation. The two attributes that correspond to the largest and second-largest numbers are chosen as the two primary attributes. For WikiPeople, since there is a primary triple for each n-ary fact in Wikidata, with its help, we simply reorganize a set of attribute-value pairs in WikiPeople to a primary triple coupled with a set of its auxiliary description(s).

The statistics of the datasets after conversion or reorganization are outlined in Table 1, where $\#Train$, $\#Valid$, and $\#Test$ are the sizes of the training set, validation set, and test set, respectively.

Dataset	$ R $	$ E $	$\#Train$	$\#Valid$	$\#Test$
JF17K	501	28,645	76,379	-	24,568
WikiPeople	193	47,765	305,725	38,223	38,281

Table 1: The statistics of the datasets.

As for metrics, we adopt the standard Mean Reciprocal Rank (MRR) and Hits@ N . For each n-ary test fact, one of its elements is removed and replaced by all the elements in E/R . These corrupted n-ary facts are fed into NeuInfer to obtain the final scores. Based on these scores, the n-ary facts are sorted in descending order, and the rank of the n-ary test fact is stored. Note that, except the n-ary test fact, other corrupted n-ary facts existing in the training/validation/test set, are discarded before sorting. This process is repeated for all other elements of the n-ary test fact. Then, MRR is the average of these reciprocal ranks, and Hits@ N is the proportion of the ranks less than or equal to N .

Knowledge inference includes entity inference and relation inference. As presented in Table 1, the number of relations and attributes in each dataset is far less than that of entities and attribute values (on JF17K, $|R| = 501$, while $|E| = 28,645$; on WikiPeople, $|R| = 193$, while $|E| = 47,765$). That is, inferring a relation/attribute is much simpler than inferring an entity/attribute value. Therefore, we adopt MRR and Hits@{1, 3, 10} on entity inference, while pouring attention to more fine-grained metrics, i.e., MRR and Hits@1 on relation inference.

5.2 Experimental Settings

The hyper-parameters of NeuInfer are tuned via grid search in the following ranges: The embedding dimension $k \in \{50, 100\}$, the batch size $\beta \in \{128, 256\}$, the learning rate $\lambda \in \{5e^{-6}, 1e^{-5}, 5e^{-5}, 1e^{-4}, 5e^{-4}, 1e^{-3}\}$, the numbers n_1 and n_2 of the neural network layers of “hrt-FCNs” and “av-FCNs” in $\{1, 2\}$, the dimension d of the interaction vector $\mathbf{o}_{hrt_{a_i}v_i}$ in $\{50, 100, 200, 400, 500, 800, 1000, 1200\}$, the weight factor w of the scores in $\{0.1, 0.2, \dots, 0.9\}$. The adopted optimal settings are: $k = 100$, $\beta = 128$, $\lambda = 5e^{-5}$, $n_1 = 2$, $n_2 = 1$, $d = 1200$, and $w = 0.1$ for JF17K; $k = 100$, $\beta = 128$, $\lambda = 1e^{-4}$, $n_1 = 1$, $n_2 = 1$, $d = 1000$, and $w = 0.3$ for WikiPeople.

5.3 Simple Knowledge Inference

Simple knowledge inference includes simple entity inference and simple relation inference. For an n-ary fact, they infer one of the entities/the relation in the primary triple or the attribute value/attribute in an auxiliary description, given its other information.

Method	JF17K				WikiPeople			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
RAE	0.310	0.219	0.334	0.504	0.172	0.102	0.182	0.320
NaLP	0.366	0.290	0.391	0.516	0.338	0.272	0.364	0.466
NeuInfer	0.517	0.436	0.553	0.675	0.350	0.282	0.381	0.467

Table 2: Experimental results of simple entity inference.

5.3.1 Baselines

Knowledge inference methods on n-ary facts are scarce. The representative methods are m-TransH (Wen et al., 2016) and its modified version RAE (Zhang et al., 2018), and the state-of-the-art one is NaLP (Guan et al., 2019). As m-TransH is worse than RAE, following NaLP, we do not adopt it as a baseline.

5.3.2 Simple Entity Inference

The experimental results of simple entity inference are reported in Table 2. From the results, it can be observed that NeuInfer performs much better than the best baseline NaLP, which verifies the superiority of NeuInfer. Specifically, on JF17K, the performance gap between NeuInfer and NaLP is significant. In essence, 0.151 on MRR, 14.6% on Hits@1, 16.2% on Hits@3, and 15.9% on Hits@10. On WikiPeople, NeuInfer also outperforms NaLP. It testifies the strength of NeuInfer treating the information in the same n-ary fact discriminatingly. By differentiating the primary triple from other auxiliary description(s), NeuInfer considers the validity of the primary triple and the compatibility between the primary triple and its auxiliary description(s) to model each n-ary fact more appropriately and reasonably. Thus, it is not surprising that NeuInfer beats the baselines. And on simpler JF17K (see Section 5.1), NeuInfer gains more significant performance improvement than on WikiPeople.

5.3.3 Simple Relation Inference

Since RAE is deliberately developed only for simple entity inference, we compare NeuInfer only with NaLP on simple relation inference. Table 3 demonstrates the experimental results of simple relation inference. From the table, we can observe that NeuInfer outperforms NaLP consistently. Detailedly, on JF17K, the performance improvement of NeuInfer on MRR and Hits@1 is 0.036 and 7.0%, respectively; on WikiPeople, they are 0.030 and 9.1%, respectively. It is ascribed to the reasonable modeling of n-ary facts, which not only improves the performance of simple entity inference but also is beneficial to pick the exact right

relations/attributes out.

Method	JF17K		WikiPeople	
	MRR	Hits@1	MRR	Hits@1
NaLP	0.825	0.762	0.735	0.595
NeuInfer	0.861	0.832	0.765	0.686

Table 3: Experimental results of simple relation inference.

5.4 Ablation Study

We perform an ablation study to look deep into the framework of NeuInfer. If we remove the compatibility evaluation component, NeuInfer is reduced to a method for binary but not n-ary facts. Since we handle knowledge inference on n-ary facts, it is inappropriate to remove this component. Thus, as an ablation, we only deactivate the validity evaluation component, denoted as NeuInfer⁻. The experimental comparison between NeuInfer and NeuInfer⁻ is illustrated in Figure 2. It can be observed from the figure that NeuInfer outperforms NeuInfer⁻ significantly. It suggests that the validity evaluation component plays a pivotal role in our method. Thus, each component of our method is necessary.

5.5 Flexible Knowledge Inference

The newly proposed flexible knowledge inference focuses on n-ary facts of arities greater than 2. It includes flexible entity inference and flexible relation inference. For an n-ary fact, they infer one of the entities/the relation in the primary triple given any number of its auxiliary description(s) or infer the attribute value/attribute in an auxiliary description given the primary triple and any number of other auxiliary description(s). In existing knowledge inference methods on n-ary facts, each n-ary fact is represented as a group of peer attributes and attribute values. These methods have not poured attention to the above flexible knowledge inference. Thus, we conduct this new type of task only on NeuInfer. Before elaborating on the experimental results, let us look into the new test set used in this section first.

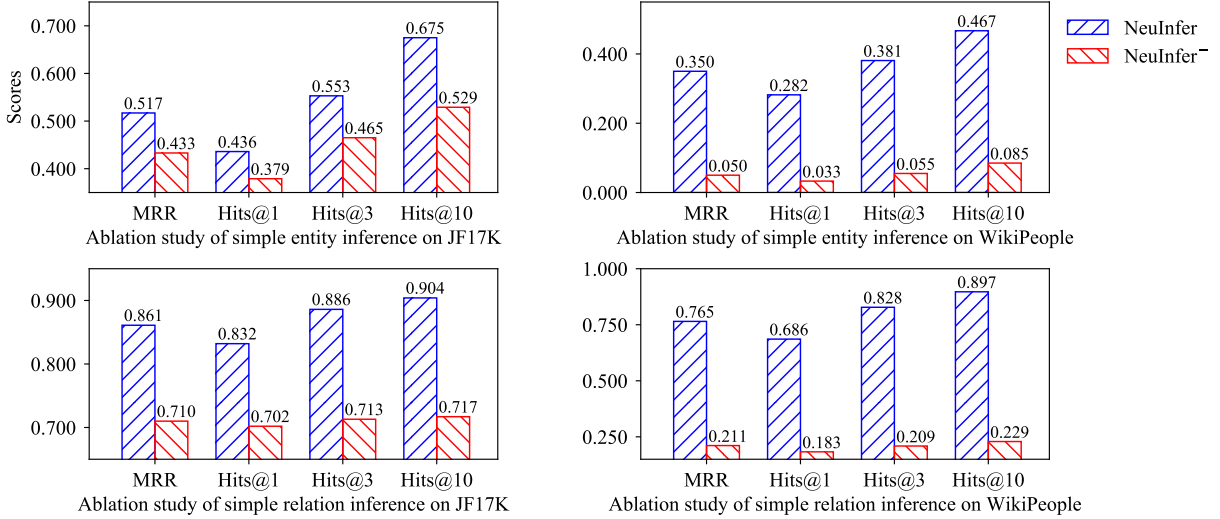


Figure 2: The experimental comparison between NeuInfer and NeuInfer⁻.

Dataset	Flexible entity inference				Flexible relation inference	
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1
JF17K	0.398	0.348	0.422	0.494	0.616	0.599
WikiPeople	0.200	0.161	0.208	0.276	0.477	0.416

Table 4: Experimental results of flexible knowledge inference.

5.5.1 The New Test Set

We generate the new test set as follows:

- Collect the n-ary facts of arities greater than 2 from the test set.
- For each collected n-ary fact, compute all the subsets of the auxiliary description(s). The primary triple and each subset form a new n-ary fact, which is added to the candidate set.
- Remove the n-ary facts that also exist in the training/validation set from the candidate set and then remove the duplicate n-ary facts. The remaining n-ary facts form the new test set.

The size of the resulting new test set on JF17K is 34,784, and that on WikiPeople is 13,833.

5.5.2 Flexible Entity and Relation Inference

The experimental results of flexible entity and relation inference on these new test sets are presented in Table 4. It can be observed that NeuInfer well tackles flexible entity and relation inference on partial facts, and achieves excellent performance. We also attribute this to the reasonable modeling of n-ary facts. For each n-ary fact, NeuInfer distinguishes the primary triple from other auxiliary description(s) and models them properly. Thus, NeuInfer well handles various types of entity and

relation inference concerning the primary triple coupled with any number of its auxiliary description(s).

5.6 Performance under Different Scenarios

To further analyze the effectiveness of the proposed NeuInfer method, we look into the breakdown of its performance on different arities, as well as on primary triples and auxiliary descriptions. Without loss of generality, here we report only the experimental results on simple entity inference.

The test sets are grouped into binary and n-ary ($n > 2$) categories according to the arities of the facts. Table 5 presents the experimental results of simple entity inference on these two categories of JF17K and WikiPeople. From the tables, we can observe that NeuInfer consistently outperforms the baselines on both categories on simpler JF17K. On more difficult WikiPeople, NeuInfer is comparable to the best baseline NaLP on the binary category and gains much better performance on the n-ary category in terms of the fine-grained MRR and Hits@1. In general, NeuInfer performs much better on JF17K than on WikiPeople. We attribute this to the simplicity of JF17K.

Where does the above performance improvement come from? Is it from inferring the head/tail entities in primary triples or the attribute values

Dataset	Method	MRR		Hits@1		Hits@3		Hits@10	
		Binary	N-ary	Binary	N-ary	Binary	N-ary	Binary	N-ary
JF17K	RAE	0.115	0.397	0.050	0.294	0.108	0.434	0.247	0.618
	NaLP	0.118	0.477	0.058	0.394	0.121	0.512	0.246	0.637
	NeuInfer	0.267	0.628	0.173	0.554	0.300	0.666	0.462	0.770
WikiPeople	RAE	0.169	0.187	0.096	0.126	0.178	0.198	0.323	0.306
	NaLP	0.351	0.283	0.291	0.187	0.374	0.322	0.465	0.471
	NeuInfer	0.350	0.349	0.278	0.303	0.385	0.364	0.473	0.439

Table 5: Experimental results of simple entity inference on binary and n-ary categories of JF17K and WikiPeople.

Dataset	Method	MRR			Hits@1			Hits@3			Hits@10		
		Binary	N-ary	Overall	Binary	N-ary	Overall	Binary	N-ary	Overall	Binary	N-ary	Overall
JF17K	NaLP	0.118	0.456	0.313	0.058	0.369	0.237	0.121	0.491	0.334	0.246	0.625	0.464
	NeuInfer	0.267	0.551	0.431	0.173	0.467	0.342	0.300	0.588	0.466	0.462	0.720	0.611
WikiPeople	NaLP	0.351	0.237	0.337	0.291	0.161	0.276	0.374	0.262	0.361	0.465	0.384	0.455
	NeuInfer	0.350	0.280	0.342	0.278	0.225	0.272	0.385	0.299	0.382	0.473	0.375	0.463

Table 6: Detailed experimental results on inferring head/tail entities.

Method	JF17K				WikiPeople			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
NaLP	0.510	0.432	0.545	0.655	0.345	0.223	0.402	0.589
NeuInfer	0.746	0.687	0.787	0.848	0.443	0.408	0.453	0.516

Table 7: Experimental results on inferring attribute values.

in auxiliary descriptions? To go deep into it, we study the performance of NeuInfer on inferring the head/tail entities and the attribute values and compare it with the best baseline NaLP. The detailed experimental results are demonstrated in Tables 6 and 7. It can be observed that NeuInfer brings more performance gain on inferring attribute values. It indicates that combining the validity of the primary triple and the compatibility between the primary triple and its auxiliary description(s) to model each n-ary fact is more effective than only considering the relatedness of attribute-value pairs in NaLP, especially for inferring attribute values.

6 Conclusions

In this paper, we distinguished the information in the same n-ary fact and represented each n-ary fact as a primary triple coupled with a set of its auxiliary description(s). We then proposed a neural network model, NeuInfer, for knowledge inference on n-ary facts. NeuInfer combines the validity evaluation of the primary triple and the compatibility evaluation of the n-ary fact to obtain the validity score of the n-ary fact. In this way, NeuInfer has the ability of well handling simple knowledge inference, which copes with the inference on whole facts. Furthermore, NeuInfer is capable of deal-

ing with the newly proposed flexible knowledge inference, which tackles the inference on partial facts consisting of a primary triple coupled with any number of its auxiliary descriptive attribute-value pair(s). Experimental results manifest the merits and superiority of NeuInfer. Particularly, on simple entity inference, NeuInfer outperforms the state-of-the-art method significantly in terms of all the metrics. NeuInfer improves the performance of Hits@3 even by 16.2% on JF17K.

In this paper, we use only n-ary facts in the datasets to conduct knowledge inference. For future works, to further improve the method, we will explore the introduction of additional information, such as rules and external texts.

Acknowledgments

The work is supported by the National Key Research and Development Program of China under grant 2016YFB1000902, the National Natural Science Foundation of China under grants U1911401, 61772501, U1836206, 91646120, and 61722211, the GFKJ Innovation Program, Beijing Academy of Artificial Intelligence (BAAI) under grant BAAI2019ZD0306, and the Lenovo-CAS Joint Lab Youth Scientist Project.

References

- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 2787–2795.
- Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. [Meta relational learning for few-shot link prediction in knowledge graphs](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4208–4217, Hong Kong, China.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 1811–1818.
- Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. [Improving knowledge graph embedding using simple constraints](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 110–121.
- Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. [Question answering over Freebase with multi-column convolutional neural networks](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 260–269.
- Takuma Ebisu and Ryutaro Ichise. 2018. TorusE: Knowledge graph embedding on a Lie group. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 1819–1826.
- Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Shared embedding based neural networks for knowledge graph completion. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 247–256.
- Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *Proceedings of the 28th International Conference on World Wide Web*, pages 583–593.
- Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. [Semantically smooth knowledge graph embedding](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 84–94.
- Prachi Jain, Pankaj Kumar, Soumen Chakrabarti, et al. 2018. [Type-sensitive knowledge base inference without explicit type supervision](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 75–80.
- Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. [Knowledge graph embedding via dynamic mapping matrix](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 687–696.
- Yantao Jia, Yuanzhuo Wang, Hailun Lin, Xiaolong Jin, and Xueqi Cheng. 2016. Locally adaptive translation for knowledge graph embedding. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 992–998.
- Xiaotian Jiang, Quan Wang, and Bin Wang. 2019. [Adaptive convolution for multi-relational learning](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 978–987, Minneapolis, Minnesota.
- Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.
- Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015a. [Modeling relation paths for representation learning of knowledge bases](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 705–714.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015b. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2181–2187.
- Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1211–1220.
- Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. [Learning attention-based embeddings for relation prediction in knowledge graphs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy.
- Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. [A novel embedding model for knowledge base completion based on convolutional neural network](#). In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–333.

- Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.
- Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816.
- Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706.
- Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2017. Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 1243–1249.
- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2071–2080.
- Denny Vrandečić and Markus Krötzsch. 2014. Wiki-data: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.
- Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1112–1119.
- Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 1300–1307.
- Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. [TransG: A generative model for knowledge graph embedding](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2316–2325.
- Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 27th International Conference on World Wide Web*, pages 1185–1194.