Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

On the Representation and Embedding of Knowledge Bases Beyond Binary Relations

Jianfeng Wen[1,2], **Jianxin Li**[1,2], **Yongyi Mao**[3], **Shini Chen**[1], and **Richong Zhang***[1,2]
[1] State Key Laboratory of Software Development Environment, Beihang University
[2] School of Computer Science and Engineering, Beihang University
[3] School of Electrical Engineering and Computer Science, University of Ottawa

Abstract

The models developed to date for knowledge base embedding are all based on the assumption that the relations contained in knowledge bases are binary. For the training and testing of these embedding models, multi-fold (or n-ary) relational data are converted to triples (e.g., in FB15K dataset) and interpreted as instances of binary relations. This paper presents a canonical representation of knowledge bases containing multi-fold relations. We show that the existing embedding models on the popular FB15K datasets correspond to a sub-optimal modelling framework, resulting in a loss of structural information. We advocate a novel modelling framework, which models multi-fold relations directly using this canonical representation. Using this framework, the existing TransH model is generalized to a new model, m-TransH. We demonstrate experimentally that m-TransH outperforms TransH by a large margin, thereby establishing a new state of the art.

1 Introduction

The emerging of knowledge bases such as YAGO[Suchanek *et al.*, 2007], DBpedia[Auer *et al.*, 2007] and Freebase[Bollacker *et al.*, 2008] has inspired intense research interest in this area, from completing and improving knowledge bases (e.g., [Baader *et al.*, 2007; Angeli and Manning, 2013]) to developing applications that retrieve information from the knowledge data (e.g., [Marin *et al.*, 2014; Xiong and Callan, 2015a; 2015b]). Recently knowledge base embedding[Bordes *et al.*, 2013; Wang *et al.*, 2014; Lin *et al.*, 2015b; Bordes *et al.*, 2014b; 2011; Socher *et al.*, 2013; Lin *et al.*, 2015a] has stood out as an appealing and generic methodology to access various research problems in this area. Briefly, this methodology sets out to represent entities in a knowledge base as points in some Euclidean space while preserving the structures of the relational data. This approach turns the discrete topology of the relations into a continuous one, enabling the design of efficient algorithms and potentially benefitting many applications. For example, embedding can be applied to link prediction [Bordes *et al.*, 2013] or question answering [Bordes *et al.*, 2014a] in knowledge bases, in which the problems are usually of a combinatorial nature in their original discrete settings.

*Corresponding author: zhangrc@act.buaa.edu.cn

Despite their promising successes, existing embedding techniques are all developed based on the assumption that knowledge data are instances of binary relations, namely instances each involving two entities (such as "Beijing is the capital of China"). In reality, however, a large portion of the knowledge data are from non-binary relations (such as "Benedict Cumberbatch played Alan Turing in the movie The Imitation Game"). For example, we observe that in Freebase[Bollacker *et al.*, 2008], more than 1/3 of the entities participate in non-binary relations. This calls for a careful investigation of embedding techniques for knowledge bases containing non-binary relations.

In this paper, we first present a clean mathematical definition of *multi-fold relations*, also known as n-ary relation in the literature [Codd, 1970]. Using this notion, we propose a canonical representation for multi-fold (binary or non-binary) relational data, which we call *instance representation*. Existing knowledge bases usually organize their data using the W3C Resource Description Framework (RDF) [Nickel *et al.*, 2015], in which relational data are represented as a collection of (subject, predicate, object) triples. Although such a triple representation, like the instance representation, is capable of capturing the structures of multi-fold relations [Codd, 1970; Nguyen *et al.*, 2014; Rouces *et al.*, 2015; Grewe, 2010; Krieger and Willms, 2015; W3C, 2016], we show that manipulating multi-fold relational data into triples (as in Freebase) results in an heterogeneity of the predicates, unfavourable for embedding. As such, we advocate that the starting point of embedding multi-fold relations should be recovering the relational data in its instance representation.

We then formulate the embedding problem on the instance representation and suggest that the heart of the problem is modelling each cost function that defines a constraint in the embedding space. We examine the popular FB15K[Bordes *et al.*, 2013] datasets, used in all existing embedding models, and point out that the triple-based data format of FB15K results from applying a particular "star-to-clique" (S2C) conversion procedure to the filtered Freebase data. This procedure can be verified to be irreversible, which causes a loss of structural information in the multi-fold relations.

Interestingly, we discover that all existing embedding models on such S2C-converted datasets can be unified under a "decomposition" modelling framework. In this framework, the cost function associated with a J-fold relation is modelled as the sum of $\binom{J}{2}$ bi-variate functions. Suggesting that the decomposition framework is fundamentally limited, we propose a "direct modelling" framework for embedding multi-fold relations. As an example in this framework, we generalize TransH[Wang *et al.*, 2014] to a new model for multi-fold relations. Although TransH is known to perform comparably to the best performing models but with lower time complexity, our experiments demonstrate that this new model, designated *m-TransH*, has even lower complexity, and outperforms TransH by an astonishing margin.

In summary, this paper takes a fundamental look at the knowledge base embedding problem when non-binary relations exist. We advocate the instance representation and the direct modelling framework on such representation. Constrained by the length requirement, we are unable to elaborate at places and certain details are omitted.

2 Multi-Fold Relations and Knowledge Base Representations

2.1 Multi-Fold Relations

A well-known algebraic concept, a binary relation [Hungerford, 2003] on a set $\mathcal{N}$ is defined as a subset of the cartesian product $\mathcal{N} \times \mathcal{N}$, or $\mathcal{N}^2$. This understanding allows an immediate generalization of binary relation to multi-fold relation (also known as n-ary relation, see, e.g. [Codd, 1970]) where $\mathcal{N}^2$ is replaced with the J-fold cartesian product $\mathcal{N}^J$ for an arbitrary integer $J \geq 2$. From a knowledge base (KB) point of view, we argue however that such algebraic definitions are incomplete, in the sense that the *role* of each coordinate in the cartesian product is not specified. For example, let R be the binary relation relating a country with its capital city. Then ambiguity exists in whether the instance "Paris is the capital of France" should be written as $(\text{Paris}, \text{France}) \in R$ or as $(\text{France}, \text{Paris}) \in R$. Although this issue is usually resolved by suitable data structures, we now formulate a clean mathematical notion of multi-fold relation which also specifies the roles of involved entities.

Throughout the paper, the following notations will be used. For any set $\mathcal{A}$ and $\mathcal{B}$, we denote by $\mathcal{B}^{\mathcal{A}}$ the set of all *functions* mapping $\mathcal{A}$ to $\mathcal{B}$, as is standard in mathematics [Hungerford, 2003]. For any function $g \in \mathcal{B}^{\mathcal{A}}$ and any subset $\mathcal{S} \subseteq \mathcal{A}$, we use $g_{:\mathcal{S}}$ to denote the restriction of function g on $\mathcal{S}$. We will use $\mathcal{N}$ to denote the set of all entities in a KB.

A multi-fold relation is then defined as follows. Let $\mathcal{M}$ be a set of *roles* in the KB, and a *multi-fold relation*, or simply, *relation*, R on $\mathcal{N}$ with roles $\mathcal{M}$ is a subset of $\mathcal{N}^{\mathcal{M}}$. Given R, we also write the set $\mathcal{M}$ as $\mathcal{M}(R)$ and call R a J-fold (or J-ary) relation if $|\mathcal{M}(R)| = J$. The value J is also referred to as the "fold" or "arity" of R. An element $t \in R$ is called an *instance* of the relation R.

Example 1 *Let R be a 3-fold relation about "which actor played which character in which movie". Then* $\mathcal{M}(R) := \{\text{ACTOR}, \text{CHARACTER}, \text{MOVIE}\}$. *The function* $t : \mathcal{M}(R) \to \mathcal{N}$ *given below is then the instance of R stating "Benedict Cumberbatch played Alan Turing in the movie The Imitation Game":*

$$\begin{array}{lcl} t(\text{ACTOR}) & = & \text{BenedictCumberbatch}, \\ t(\text{CHARACTER}) & = & \text{AlanTuring}, \\ t(\text{MOVIE}) & = & \text{TheImitationGame}. \end{array}$$

2.2 Instance Representation

Let $\mathcal{R}$ index a set of distinct multi-fold relations on $\mathcal{N}$. More precisely put, for each $r \in \mathcal{R}$, there is a relation R_r on $\mathcal{N}$ with roles $\mathcal{M}(R_r)$. One may identify the index r of R_r with the *type* of the relation R_r, and in this view, $\mathcal{R}$ is a collection of *relation types*. Let $\mathcal{T}_r$ be the set of instances of relation R_r that are included in the KB, then the KB can be specified as $(\mathcal{N}, \mathcal{R}, \{\mathcal{T}_r, r \in \mathcal{R}\})$. We call such specification an *instance representation*. Since a KB is usually incomplete, each set $\mathcal{T}_r$ is expected to be strictly contained in R_r. As a consequence, relation R_r is in fact unknown, and all information about R_r is revealed via the set $\mathcal{T}_r$, sampled from R_r. Clearly, the instance representation contains all information in the KB pertaining to the structures of the relations and how they interact. Therefore, instance representations are legitimately canonical, at least from the embedding perspective, where only such information matters.

2.3 Fact Representation

For various implementation considerations, practical KBs such as Freebase [Fre, 2016] organize relational data in a different format, the core of which is a notion related to but different from the multi-fold relation we define. We call this notion *meta-relation* and define it next.

Given a set $\mathcal{N}$ of entities and a set $\mathcal{M}$ of roles, a *(multi-fold) meta-relation* Q on $\mathcal{N}$ with roles $\mathcal{M}$ is a subset of $\left(2^{\mathcal{N}}\right)^{\mathcal{M}}$, where $2^{\mathcal{N}}$ is the power set of $\mathcal{N}$. That is, each element in the meta-relation Q, which will be called a *fact* of Q, is a function mapping $\mathcal{M}$ to $2^{\mathcal{N}}$. Similar to relations, we call Q a J-fold meta-relation if $|\mathcal{M}| = J$ and often write $\mathcal{M}(Q)$ in place of $\mathcal{M}$.

Example 2 *Let Q be a 3-fold meta-relation about "who played what instruments in the recording of what music piece". The roles $\mathcal{M}(Q)$ consists of* RECORDING *(i.e., the music piece),* INSTRUMENT_ROLE *(i.e., the music instrument), and* CONTRIBUTOR *(i.e., the person).*

Let u_1 and u_2 be two facts of Q, which respectively state "Will McGregor played bass and bass guitar in Precious Things" and "Michael Harrison played violin in Pretty Good Year". As functions in $(2^{\mathcal{N}})^{\mathcal{M}(Q)}$, u_1 and u_2 are given in Table 1.

For any given meta-relation Q, the number of entities involved in a fact $u \in Q$ may vary with u in general. Recall that in a J-fold relation, each instance involves exactly J entities. For later reference (in Section 2.6), a meta-relation Q is said to be *degenerate* if every $u \in Q$ is such that $u(\rho)$ is a singleton set for every $\rho \in \mathcal{M}(Q)$.

Parallel to instance representations, one can similarly define the *fact representation* $(\mathcal{N}, \mathcal{Q}, \{\mathcal{U}_q : q \in \mathcal{Q}\})$ for a KB: $\mathcal{Q}$ indexes a set $\{Q_q : q \in \mathcal{Q}\}$ of distinct meta-relations where each Q_q is a meta-relation on $\mathcal{N}$ with roles $\mathcal{M}(Q_q)$; and each $\mathcal{U}_q$ is a set of facts of Q_q.

2.4 Graphical Representations

Both instance representations and fact representations can be associated with an edge-labelled bi-partite graph. For an instance (resp. fact) representation, its associated graph contains two sets of vertices, representing entities and instances (resp. facts) respectively; if an entity is involved in an instance (resp. fact), the corresponding entity vertex is connected to corresponding instance (resp. fact) vertex by an edge, and the edge label is the role of the entity in the instance (resp. fact). The reader is referred to Figures 1 and 2 for examples of such graphical notations. As such, we may sometimes use a graph-theoretic language when speaking of instance or fact representations.

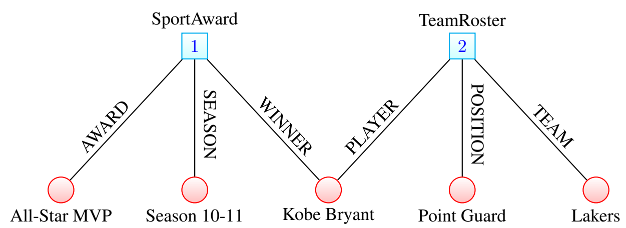

Figure 1: The bipartite graph for a toy instance representation containing two instances. Instance 1 is from the SportAward relation (or having relation type SportAward), stating "Kobe Bryant is the All-Star MVP for the season 2010-2011". Instance 2 is from the TeamRoster relation, stating "Kobe Bryant is a Point Guard in Los Angeles Lakers".

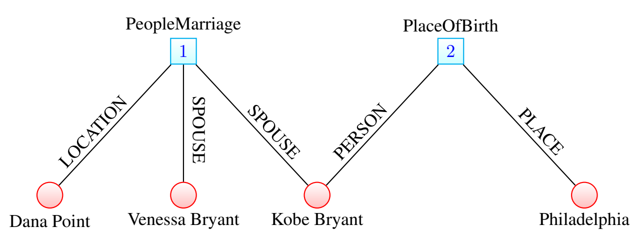

Figure 2: The bipartite graph for a toy fact representation containing two facts. Fact 1 is from the PeopleMarriage meta-relation (noting that two entities have the same role SPOUSE), stating "Kobe Bryant and Venessa Bryant became married in Dana Point". Fact 2 is from the PlaceOfBirth meta-relation, stating "Kobe Bryant was born in Philadelphia" (noting that such a fact is also an instance).

2.5 Converting Facts to Instances

We now show that a fact representation $\mathcal{F} := (\mathcal{N}, \mathcal{Q}, \{\mathcal{U}_q : q \in \mathcal{Q}\})$ can be converted to an instance representation.

Let $\mathcal{N}' := \mathcal{N} \cup \left(\bigcup_{q\in\mathcal{Q}} \mathcal{U}_q\right)$ and let $\mathcal{M}'(Q) := \mathcal{M}(Q) \cup \{\text{FACT–ID}\}$ for any arbitrary meta-relation Q on $\mathcal{N}$. That is, the ID (or name) of each fact in $\mathcal{F}$ is regarded as an "entity" and an additional "role" FACT–ID is generated. For any meta-relation Q on $\mathcal{N}$ and a fact $u \in Q$, let $T_{\text{id}}(u)$ be the set of all distinct functions in $\mathcal{N}'^{\mathcal{M}'(Q)}$ in which each function t maps a role $\rho \in \mathcal{M}(Q)$ to an entity in $u(\rho)$ and maps FACT–ID to the ID of the fact u. It is easy to see that $T_{\text{id}}(u)$ contains precisely $\left|\prod_{\rho\in\mathcal{M}(Q)} u(\rho)\right|$ functions. For any collection $\mathcal{U}$ of facts, let $T_{\text{id}}(\mathcal{U}) := \bigcup_{u\in\mathcal{U}} T_{\text{id}}(u)$. Then for any meta-relation Q, $T_{\text{id}}(Q)$ is a subset of $\mathcal{N}'^{\mathcal{M}'(Q)}$, namely a relation on $\mathcal{N}'$ with roles $\mathcal{M}'(Q)$.

Denote $T_{\text{id}}(\mathcal{F}) := (\mathcal{N}', \mathcal{Q}, \{T_{\text{id}}(\mathcal{U}_q) : q \in \mathcal{Q}\})$. Clearly, $T_{\text{id}}(\mathcal{F})$ is the instance representation of a KB "augmented" from $\mathcal{F}$, where fact ID's in $\mathcal{F}$ are added to the entities, and FACT–ID is taken as an additional role.

Lemma 1 *$\mathcal{F}$ can be recovered from $T_{\text{id}}(\mathcal{F})$.*

If one only wishes for an instance representation of $\mathcal{F}$ without demanding the recoverability in Lemma 1, a simpler conversion can be defined: for a fact $u \in Q$, $T(u) := \{t_{:\mathcal{M}(Q)} : t \in T_{\text{id}}(u)\}$. That is, $T(u)$ is similar to T_{id} but having fact ID dropped. Similarly for any collection $\mathcal{U}$ of facts, let $T(\mathcal{U}) := \bigcup_{u\in\mathcal{U}} T(u)$. Then for any meta-relation Q, $T(Q)$ is a relation on $\mathcal{N}$ with roles $\mathcal{M}(Q)$. Denote $T(\mathcal{F}) := (\mathcal{N}, \mathcal{Q}, \{T(\mathcal{U}_q) : q \in \mathcal{Q}\})$. Then $T(\mathcal{F})$ is an instance representation for $\mathcal{F}$, but $\mathcal{F}$ can not be recovered from $T(\mathcal{F})$ in general.

The reader is referred to Table 1 for an example converting a fact to instances using $T_{\text{id}}(\cdot)$ and $T(\cdot)$. Such conversions may also be viewed graphically as shown in Figure 3. Later in our experiments, we will investigate, relative to $T(\mathcal{F})$, whether the fact-level information contained in $T_{\text{id}}(\mathcal{F})$ (namely, that certain instances belong to the same fact) is useful for embedding.

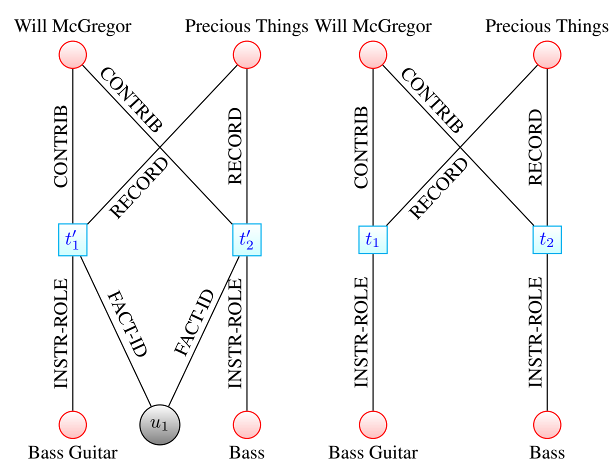

Figure 3: Fact u_1 in Table 1 (or Example 2) converted to instances using T_{id} (left) and T (right) respectively.

2.6 Freebase

Freebase [Fre, 2016] is a large collaborative KB. The core of Freebase may be understood as a fact representation, con-

Table 1: The facts u_1 and u_2 in Example 2 and conversions of u_1 to instances: $T(u_1) := \{t_1, t_2\}$, $T_{\text{id}}(u_1) := \{t'_1, t'_2\}$.

role	u_1(role)	u_2(role)	t_1(role)	t_2(role)	t'_1(role)	t'_2(role)
CONTRIBUTOR	{WillMcGregor}	{MichaelHarrison}	WillMcGregor	WillMcGregor	WillMcGregor	WillMcGregor
RECORDING	{PreciousThings}	{PrettyGoodYear}	PreciousThings	PreciousThings	PreciousThings	PreciousThings
INSTRUMENT–ROLE	{BassGuitar, Bass}	{Violin}	BassGuitar	Bass	BassGuitar	Bass
FACT–ID					u_1	u_1

taining about 3 billion facts and involving 50 million entities. However, following the RDF [Nickel *et al.*, 2015] format, Freebase organizes its data as a collection of *triples*, which deviates from the fact representation defined in this paper. Let $\mathcal{F}$ denote the underlying fact representation of Freebase. The essential format of Freebase can be obtained by manipulating $\mathcal{F}$ as follows.

For each fact in a binary degenerate meta-relation (defined in Section 2.3), apply the S2C conversion (Figure 4) to the fact vertex. This results in a collection of entity-predicate-entity triples, which are equivalent to instances of binary relations. The remaining fact vertices are called CVT (Common Value Type) vertices in Freebase. Each labelled edge connected to a CVT vertex is represented as an entity-role-CVT triple. For each meta-relation Q_q that is not binary degenerate, Freebase introduces a Mediator vertex to indicate the type q of Q_q; then for each fact $u \in Q_q$, a Mediator-CONTAINS-CVT triple is created, where CVT indicates u and Mediator indicates q, and CONTAINS is a fixed global token, independent of u and q, serving to mean $u \in Q_q$.

Using the three kinds of triples, Freebase's data organization is equivalent to the fact representation $\mathcal{F}$. However, the heterogeneity in the triple semantics makes this representation not as clean as the instance or fact representation, at least for embedding purpose.

Star-to-Clique (S2C) Conversion

On an edge labelled graph $\mathcal{G}$, let (u, r, v) denote the labelled edge connecting vertices u and v with label r, and let $\mathcal{N}(v)$ denote the set of all adjacent vertices of v. Then the star-to-clique conversion on a vertex s is defined by the following procedure.
(1) For every $x_1, x_2 \in \mathcal{N}(s)$ forming two labelled edges (x_1, r_1, s) and (x_2, r_2, s) where $r_1 \neq r_2$, add a labelled edge $(x_1, r_1.r_2, x_2)$ to $\mathcal{G}$.
(2) Delete s and all edges connecting to s.

The reader is referred to Figure 5 for an example.

Figure 4: Definition of Star-to-Clique Conversion

3 Embedding

3.1 Problem Formulation

Instance representations, fact representations, and the RDF-based triple representations (e.g., that in Freebase) may all contain equivalent structural information about the relations

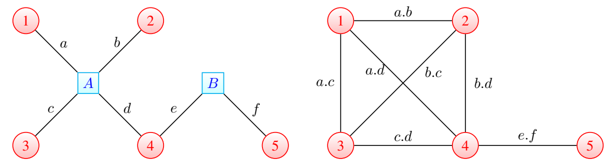

Figure 5: Applying the S2C conversion to vertices A and B in the labelled graph on the left results in the labelled graph on the right.

contained in a KB. However, instance representations have a uniform semantics comparing with the RDF-based triple representations and are easier to deal with than fact representations. For this reason, we now formulate the KB embedding problem on an instance representation $(\mathcal{N}, \mathcal{R}, \{\mathcal{T}_r : r \in \mathcal{R}\})$.

Let a vector space U over the field $\mathbb{R}$ of real numbers be the chosen space for embedding. The objective of KB embedding is to construct a function $\phi : \mathcal{N} \to U$ and a subset $C_r \subset U^{\mathcal{M}(R_r)}$ for each relation R_r such that *ideally* the following properties are satisfied.

1. For every $r \in \mathcal{R}$ and every instance $t \in R_r$, $\phi \circ t \in C_r$, where the symbol $\circ$ denotes function composition [Hungerford, 2003].

2. For every $r \in \mathcal{R}$ and every function $t \in \mathcal{N}^{\mathcal{M}(R_r)} \setminus R_r$, $\phi \circ t \notin C_r$.

Here, the function ϕ, serving as a representation of $\mathcal{N}$, maps an entity to its embedding vector. The subsets $\{C_r : r \in \mathcal{R}\}$, serving as a representation of $\{R_r : r \in \mathcal{R}\}$, define a set of constraints on the embedding vectors which preserve the intra-relational and inter-relational structures of $\{R_r : r \in \mathcal{R}\}$.

Note that each constraint C_r may be identified with a non-negative cost function $f_r : U^{\mathcal{M}(R_r)} \to \mathbb{R}$ such that

$$f_r(\mathbf{t}) = 0 \quad \text{if} \quad \mathbf{t} \in C_r, \text{ and} \tag{1}$$

$$f_r(\mathbf{t}) > 0 \quad \text{if} \quad \mathbf{t} \notin C_r \tag{2}$$

Denote $\Theta := \{f_r : r \in \mathcal{R}\}$. The problem then translates to determining (Θ, ϕ). But $\{R_r : r \in \mathcal{R}\}$ is unknown, and all we have is the observed instances $\{\mathcal{T}_r : r \in \mathcal{R}\}$ and possibly some "negative examples" $\{\mathcal{T}_r^- : r \in \mathcal{R}\}$, where each $\mathcal{T}_r^- \subset \mathcal{N}^{\mathcal{M}(R_r)} \setminus R_r$. Note that when the KB is large, for any $t \in \mathcal{T}_r$, if we replace its value $t(\rho)$ for some role $\rho \in \mathcal{M}(R_r)$ with a random entity, the resulting function falls in $\mathcal{N}^{\mathcal{M}(R_r)} \setminus R_r$ with high probability. This can be used to construct $\mathcal{T}_r^-$ ([Bordes *et al.*, 2013; Wang *et al.*, 2014; Lin *et al.*, 2015b]).

Treating the problem as *learning* (Θ, ϕ), we may not need the property 1 above to hold strictly. Then the equality "$= 0$"

in (1) is taken as "as close to 0 as possible". Towards a margin-based optimization formulation (which gives better discriminative power and robustness), the threshold 0 in (2) is raised to a positive value c. The problem can then be formulated as finding (Θ, ϕ) to minimize the following global cost function.

$$F(\Theta, \phi) := \sum_{r \in \mathcal{R}} \left(\sum_{t \in \mathcal{T}_r} f_r(\phi \circ t) + \sum_{t^- \in \mathcal{T}_r^-} \left[c - f_r(\phi \circ t^-) \right]_+ \right), \tag{3}$$

where $[\cdot]_+$ denotes the rectifier function [Glorot *et al.*, 2011], namely, $[a]_+ := \max(0, a)$.

What remains is to choose a proper space of Θ for this optimization problem, which is at the heart of modelling.

3.2 FB15K Datasets

A popular dataset for training and testing embedding models is the FB15K datasets [Bordes *et al.*, 2013], filtered from Freebase. When viewing Freebase as an edge-labelled graph (i.e., every triple as an labelled edge), we observe that to every CVT vertex in the filtered data, FB15K has applied the S2C conversion (Figure 4).

Lemma 2 *After applying S2C conversions to a graph $\mathcal{G}$, in general, $\mathcal{G}$ is no longer recoverable.*

This suggests a loss of structural information in this conversion. Working with this dataset, for which the original fact or instance representation is no longer recoverable, one is left with no option for embedding multi-fold relations but to treat the triples as instances of binary relations. In addition, we observe that the S2C conversions applied in FB15K have also involved the Mediator vertices connected to the CVTs. This results in each Mediator vertex in FB15K connecting to a good number of entity vertices. This connectivity provides no information about the structures of the relations, arguably only serving as "noise" for embedding. For these reasons and since we want to work with multi-fold relational data in their intact form, FB15K no longer suits our purpose.

3.3 Prior Art of Modelling

To date, well-known models developed for KB embedding include TransE [Bordes *et al.*, 2013], TransH[Wang *et al.*, 2014], TransR [Lin *et al.*, 2015b], the Unstructured Model (UE) [Bordes *et al.*, 2014b], the Structure Embedding Model [Bordes *et al.*, 2011], the Neural Tensor Network model[Socher *et al.*, 2013] and the Single Layer Model[Socher *et al.*, 2013]. All these models deal with datasets in a triple representation (such as FB15K) and treat each triple as an instance of a binary relation. This treatment is equivalent to regarding the triple representation as an instance representation where each relation R_r is binary. Consequently, each cost function f_r may be regarded as a function on U^2. For example, in TransH, the function f_r is defined by

$$f_r(\mathbf{x}, \mathbf{y}) = \|\mathbb{P}_{\mathbf{n}_r}(\mathbf{x}) + \mathbf{d}_r - \mathbb{P}_{\mathbf{n}_r}(\mathbf{y})\|^2, \tag{4}$$

where $\mathbf{n}_r$ a unit-length vector in U, $\mathbf{d}_r$ is a vector in the hyperplane in U with normal vector $\mathbf{n}_r$, and $\mathbb{P}_{\mathbf{n}_r} : U \to U$ is the function that maps a $\mathbf{z} \in U$ to the projection of $\mathbf{z}$ on the hyperplane with normal vector $\mathbf{n}_r$, namely,

$$\mathbb{P}_{\mathbf{n}_r}(\mathbf{z}) := \mathbf{z} - \mathbf{z}^T \mathbf{n}_r \mathbf{n}_r.$$

Among these models, TransE is arguably the most influential, which has inspired the later models in the "Trans series". TransR is reportedly the best performing model [Lin *et al.*, 2015b]. TransH slightly under-performs TransR on FB15K, but having much lower computation complexity.

Decomposition Framework: Let $\mathcal{G} := (\mathcal{N}, \mathcal{R}, \{\mathcal{T}_r : r \in \mathcal{R}\})$ be an instance representation. Let Γ_r be the set of all size-2 subsets of $\mathcal{M}(R_r)$. In the decomposition framework, every $f_r : U^{\mathcal{M}(R_r)} \to \mathbb{R}$ is parametrized as

$$f_r(\mathbf{t}) := \sum_{\gamma \in \Gamma_r} f_\gamma(\mathbf{t}_{:\gamma}). \tag{5}$$

Let $\mathrm{S2C}(\mathcal{G})$ denote the triple representation resulting from S2C-converting every instance vertex in $\mathcal{G}$.

Lemma 3 *The global cost function (3) for $\mathrm{S2C}(\mathcal{G})$ is equivalent to the global function (3) for $\mathcal{G}$ under the parametrization (5).*

That is, models on a triple representation resulting from S2C conversion of an instance representation are equivalent to modeling the original instance representation using the decomposition framework. Noting that the FB15K data have undergone the S2C conversion, all afore-mentioned models on FB15K are, coincidentally, special cases of the decomposition framework. Since the S2C conversion distorts the structures of the relations in the KB (Lemma 2), the decomposition framework necessarily suffers from a loss of information.

In addition, it is possible to show that parametrization using the decomposition framework can result in large errors in approximating the function f_r.

Recently another model PTransE[Lin *et al.*, 2015a] has been proposed. An extension of TransE without considering multi-fold relations, PTransE performs comparably to TransR.

3.4 Proposed Model

As the decomposition framework is fundamentally limited, we advocate a *direct modelling framework*, namely, that the cost function f_r is modelled directly without recourse to the decomposition in (5). We now present an example model in this framework, termed *m-TransH*, which generalizes TransH directly to multi-fold relations.

In *m-TransH*, each cost function f_r is parametrized by two unit-length orthogonal vectors $\mathbf{n}_r$ and $\mathbf{b}_r$ in U and a function $\mathbf{a}_r \in \mathbb{R}^{\mathcal{M}(R_r)}$. More specifically, the function f_r is defined by

$$f_r(\mathbf{t}) := \left\| \sum_{\rho \in \mathcal{M}(R_r)} \mathbf{a}_r(\rho) \mathbb{P}_{\mathbf{n}_r}(\mathbf{t}(\rho)) + \mathbf{b}_r \right\|^2, \ \mathbf{t} \in \mathcal{N}^{\mathcal{M}(R_r)}. \tag{6}$$

In addition, the orthogonality and unit-lengths constraints on the parameters $\mathbf{b}_r$ and $\mathbf{n}_r$ is implemented as L2 penalizing terms added to the cost function f_r in (6).

Lemma 4 *If each R_r is a binary relation and $\sum_{\rho \in \mathcal{M}(R_r)} \mathbf{a}_r(\rho) = 0$, the optimization problem of m-TransH (specified via (3) and (6)) and that of TransH (specified via (3) and (4)) are identical.*

Thus m-TransH reduces to TransH for binary relations.

4 Experiments

4.1 JF17K Datasets

The full Freebase data in RDF format was downloaded. Entities involved in very few triples and the triples involving *String*, *Enumeration Type* and *Numbers* were removed. A fact representation was recovered from the remaining triples. Facts from meta-relations having only a single role were removed. From each meta-relation containing more than 10000 facts, 10000 facts were randomly selected. Denote the resulting fact representation by $\mathcal{F}$. Two instance representations[1] $T_{\mathrm{id}}(\mathcal{F})$ and $T(\mathcal{F})$ were constructed. Further filtering was applied to $T(\mathcal{F})$ such that each entity is involved in at least 5 instances. Denote the filtered $T(\mathcal{F})$ by $\mathcal{G}$. Then $T_{\mathrm{id}}(\mathcal{F})$ was filtered correspondingly so that it contains the same set of instances as $\mathcal{G}$. Denote the filtered $T_{\mathrm{id}}(\mathcal{F})$ by $\mathcal{G}_{\mathrm{id}}$. Instance representation $\mathrm{S2C}(\mathcal{G})$ was constructed and is denoted by $\mathcal{G}_{\mathrm{s2c}}$. This resulted in three consistent datasets, $\mathcal{G}$, $\mathcal{G}_{\mathrm{id}}$ and $\mathcal{G}_{\mathrm{s2c}}$.

Dataset $\mathcal{G}_{\mathrm{id}}$ was randomly split into training set $\mathcal{G}_{\mathrm{id}}^{\checkmark}$ and testing set $\mathcal{G}_{\mathrm{id}}^{?}$ where every fact ID entity in $\mathcal{G}_{\mathrm{id}}^{?}$ was assured to appear in $\mathcal{G}_{\mathrm{id}}^{\checkmark}$. The corresponding splitting was then applied to $\mathcal{G}$ and $\mathcal{G}_{\mathrm{s2c}}$, giving rise to training sets $\mathcal{G}^{\checkmark}$ and $\mathcal{G}_{\mathrm{s2c}}^{\checkmark}$, and testing sets $\mathcal{G}^{?}$ and $\mathcal{G}_{\mathrm{s2c}}^{?}$. We call these datasets JF17K. Their statistics, in the same order as FB15K, are given in Table 2.

Table 2: Statistics of JF17K.

	$\mathcal{G}^{\checkmark}/\mathcal{G}_{\mathrm{id}}^{\checkmark}$	$\mathcal{G}_{\mathrm{s2c}}^{\checkmark}$	$\mathcal{G}^{?}/\mathcal{G}_{\mathrm{id}}^{?}$	$\mathcal{G}_{\mathrm{s2c}}^{?}$
# of entities	17629	17629	12282	12282
# of instances/triple types	181	381	159	336
# of instances/triples	139997	254366	22076	52933

4.2 Training and Testing

We performed four kinds of experiments, termed `m-TransH`, `m-TransH:ID`, `TransH:triple` and `TransH:inst`, in which `m-TransH` and `m-TransH:ID` train an m-TransH model, and `TransH:triple` and `TransH:inst` train a TransH model. The training and testing datasets for the experiments are given in Table 3.

Table 3: The training and testing datasets for the experiments.

m-TransH	m-TransH:ID	TransH:triple	TransH:inst
$(\mathcal{G}^{\checkmark}, \mathcal{G}^{?})$	$(\mathcal{G}_{\mathrm{id}}^{\checkmark}, \mathcal{G}_{\mathrm{id}}^{?})$	$(\mathcal{G}_{\mathrm{s2c}}^{\checkmark}, \mathcal{G}_{\mathrm{s2c}}^{?})$	$(\mathcal{G}_{\mathrm{s2c}}^{\checkmark}, \mathcal{G}_{\mathrm{s2c}}^{?})$

In `TransH:triple` and `TransH:inst`, for each triple in $\mathcal{G}_{\mathrm{sc}}^{\checkmark}$, one random negative example is generated. In `m-TransH` and `m-TransH:ID`, for each instance in $\mathcal{G}^{\checkmark}$, $\binom{L}{2}$ random negative examples are generated. This way, the total number of negative examples used in every experiment is the same, assuring a fair comparison. Stochastic Gradient Descent is used for training, as is standard. Several choices of the dimension (DIM) of U are studied.

Testing Protocol: In `m-TransH`, `m-TransH:ID` and `TransH:triple`, for each testing instance/triple and for each entity x therein, assume that x is unknown; evaluate the cost function for the embedding of the instance/triple by replacing x with every $x' \in \mathcal{N}$, rank the cost of x' from low to high, and record the rank for $x' = x$. Hit@10 (HIT) and Mean Rank (RANK) are used as the performance metrics[Bordes *et al.*, 2013]. Since the number of triples in $\mathcal{G}_{\mathrm{s2c}}^{?}$ is significantly larger than the number of instances in $\mathcal{G}_{\mathrm{id}}^{?}$ and $\mathcal{G}^{?}$, it is questionable whether this testing protocol is fair to the TransH model used in `TransH:triple`. To assure fairness, in `TransH:inst`, the TransH model is interpreted as parametrizing the cost function for each relation type in $\mathcal{G}^{?}$ using the decomposition framework (5); then for each instance in $\mathcal{G}^{?}$, each entity x therein is queried only once in `TransH:inst`, and the cost for each replacing entity is computed at the instance level instead, using (5) and (4).

4.3 Results and Discussions

The overall performances of the four kinds of experiments are shown in Table 4 and Figure 6.

With the two different testing protocols, TransH performs similarly, where the achieved HIT values are virtually identical and the instance-level test appears somewhat inferior in RANK. This suggests that using the triple-level test, `TransH:triple`, to compare with m-TransH at least won't depreciate TransH. Comparing with its reported performance on FB15K [Wang *et al.*, 2014], TransH performs rather significantly better on JF17K. This, to an extent attributed to the intrinsic difference in the data, may also reflect that the filtering process of JF17K is more delicate. For example, JF17K contains no Mediator vertices, hence "cleaner".

Overall m-TransH on both $\mathcal{G}^{?}$ and $\mathcal{G}_{\mathrm{id}}^{?}$ outperforms TransH by a huge margin. We believe that this performance gain is solely due to the fact that multi-fold relations are treated properly with the direct modelling framework. It is pleasant to see that HIT above 80% can be achieved in this framework, significantly exceeding all reported embedding performances (which are about 50%). This leap perhaps signals that KB embedding is at the verge of prevailing in practice. The observation that `m-TransH:ID` consistently outperforms `m-TransH` in HIT implies that fact-level information can be useful for embedding, a direction certainly deserving further investigation. However, the RANK positions of two m-TransH settings are reversed: `m-TransH:ID` underperforms `m-TransH`, and the RANK gap decreases as DIM increases. This is because a large number (36201) of fact IDs, twice the number of entities, need to be embedded in `m-TransH:ID`. When DIM is low, there are not sufficient degrees of freedom to fit all the entities and the fact IDs. This makes a fraction of entities embedded particularly poorly and brings the overall RANK worse than `m-TransH`. As DIM increases, this problem becomes less severe, resulting in improved RANK.

[1]In the construction of $T_{\mathrm{id}}(\mathcal{F})$, when acting on facts of degenerate meta-relations, T_{id} is taken as T. Such facts each contain a single instance, and there is no need to keep their ID's.

Table 4: Overall performances (HIT/RANK)

DIM	TransH:triple	TransH:inst	m-TransH	m-TransH:ID
25	53.12%/**74.6**	52.64%/**79.0**	63.45%/67.8	72.12%/84.5
50	**54.21**%/75.0	**53.10**%/79.2	65.87%/60.8	72.54%/78.5
100	53.32%/78.9	52.86%/82.7	67.54%/59.4	78.61%/70.5
150	52.11%/85.4	51.91%/89.7	68.12%/60.0	77.51%/63.3
200	50.54%/87.8	50.98%/91.1	**68.75**%/59.8	77.81%/64.7
250	49.26%/92.4	50.10%/95.9	68.73%/**58.7**	**80.82**%/**61.4**

Table 5: Breakdown performances (HIT/RANK) across relations with different fold J.

J	TransH:triple DIM=50	TransH:inst DIM=50	m-TransH DIM=250	m-TransH:ID DIM=250
2	50.29%/102.7	50.30%/102.4	58.60%/78.7	65.96%/99.3
3	49.99%/86.6	50.79%/77.5	67.73%/60.3	90.38%/39.2
4	56.58%/56.2	59.69%/43.1	90.19%/18.5	92.10%/32.2
5	75.93%/8.1	76.13%/7.0	93.93%/4.7	95.36%/5.8
6	98.52%/7.8	100%/1.5	100%/1.5	100%/2.0

For each kind of experiments, we select the dimension (DIM) at which it performs the best and break down its performance across relations with different fold J. We observe in Table 5 that for *every* fold value J, m-TransH outperforms TransH. Note that m-TransH is designed to handle multi-fold relations. However its improved embedding performance for non-binary relations clearly also has a global impact, allowing the binary relations to be embedded better as well.

Finally, we note that the time complexity of m-TransH is significantly lower than TransH. For example, at DIM=50, the training/testing times (in minutes) for `TransH:triple` and `m-TransH:ID` are respectively 105/229 and 52/135, on a 32-core Intel E5-2650 2.0GHz processor. This is because with the decomposition framework, an instance of a J-fold relation is S2C-converted to $\binom{J}{2}$ triples, greatly increasing the number of model parameters.

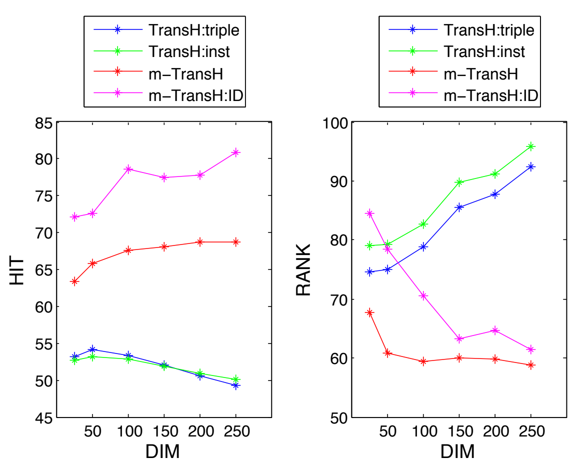

Figure 6: Overall performances in Table 4.

5 Concluding Remarks

This paper examines the fundamentals of multi-fold relations and advocates instance representations as a canonical representation for knowledge bases. We show that the widely adopted decomposition framework, which models multi-fold relations at the triple level, is fundamentally limited. Instead, we propose to model multi-fold relations at the instance level. With a simple example of such models, we demonstrate great advantages of this approach both in performance and in complexity. Outperforming TransH by a cheerful margin, this simple model, m-TransH, perhaps signals the arrival of a new performance regime in knowledge base embedding.

To inspire further research on the embedding of multi-fold relations, we have made our JF17K datasets publicly available.[2]

6 Acknowledgement

This work is supported partly by China 973 program (No. 2014CB340305), partly by the National Natural Science Foundation of China (No. 61300070, 61421003), and partly by the Beijing Advanced Innovation Center for Big Data and Brain Computing.

References

[Angeli and Manning, 2013] Gabor Angeli and Christopher D Manning. Philosophers are mortal: Inferring the truth of unseen facts. In *CoNLL*, pages 133–142, 2013.

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[Baader *et al.*, 2007] Franz Baader, Bernhard Ganter, Baris Sertkaya, and Ulrike Sattler. Completing description logic knowledge bases using formal concept analysis. In *IJCAI*, volume 7, pages 230–235, 2007.

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, number EPFL-CONF-192344, 2011.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[Bordes *et al.*, 2014a] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014.

[2] http://www.site.uottawa.ca/~yymao/JF17K

[Bordes *et al.*, 2014b] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014.

[Codd, 1970] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[Fre, 2016] Freebase. http://www.freebase.com, 2016. Accessed: 2016-01-20.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[Grewe, 2010] N Grewe. A generic reification strategy for n-ary relations in dl. In H. Herre et al., editors, *Proceedings of the 2nd workshop of the GI-Fachgruppe 'Ontologien in Biomedizin und Lebenswissenschaften' (OBML) : Mannheim, Germany*, 2010.

[Hungerford, 2003] Thomas Hungerford. *Algebra*. Springer, eighth edition, 2003.

[Krieger and Willms, 2015] Hans-Ulrich Krieger and Christian Willms. Extending owl ontologies by cartesian types to represent n-ary relations in natural language. In *Language and Ontologies 2015. Language and Ontologies, located at 11th International Conference on Computational Semantics, April 14, London, United Kingdom*. o.A., 4 2015.

[Lin *et al.*, 2015a] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*, 2015.

[Lin *et al.*, 2015b] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, 2015.

[Marin *et al.*, 2014] Alex Marin, Roman Holenstein, Ruhi Sarikaya, and Mari Ostendorf. Learning phrase patterns for text classification using a knowledge graph and unlabeled data. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[Nguyen *et al.*, 2014] Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like rdf reification?: Making statements about statements using singleton property. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 759–770, New York, NY, USA, 2014. ACM.

[Nickel *et al.*, 2015] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *arXiv preprint arXiv:1503.00759*, 2015.

[Rouces *et al.*, 2015] Jacobo Rouces, Gerard de Melo, and Katja Hose. Framebase: Representing n-ary relations using semantic frames. In *The Semantic Web. Latest Advances and New Domains*, pages 505–521. Springer, 2015.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

[Suchanek *et al.*, 2007] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[W3C, 2016] Defining N-ary Relations on the Semantic Web. http://www.w3.org/TR/swbp-n-aryRelations/, 2016. Accessed: 2016-01-20.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119. Citeseer, 2014.

[Xiong and Callan, 2015a] Chenyan Xiong and Jamie Callan. Esdrank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 951–960, New York, NY, USA, 2015. ACM.

[Xiong and Callan, 2015b] Chenyan Xiong and Jamie Callan. Query expansion with Freebase. In *Proceedings of the Fifth ACM International Conference on the Theory of Information Retrieval*. ACM, 2015.