Knowledge Graph Embedding with Hierarchical Relation Structure

Zhao Zhang[1,2], **Fuzhen Zhuang**[1,2*], **Meng Qu**[3], **Fen Lin**[4], **Qing He**[1,2]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Rutgers Business School, Rutgers University, New Jersey, 07102, USA
[4]Search Product Center, WeChat Search Application Department, Tencent, China
{zhangzhao2017, zhuangfuzhen, heqing}@ict.ac.cn
∗corresponding author

Abstract

The rapid development of knowledge graphs (KGs), such as Freebase and WordNet, has changed the paradigm for AI-related applications. However, even though these KGs are impressively large, most of them are suffering from incompleteness, which leads to performance degradation of AI applications. Most existing researches are focusing on knowledge graph embedding (KGE) models. Nevertheless, those models simply embed entities and relations into latent vectors without leveraging the rich information from the relation structure. Indeed, relations in KGs conform to a three-layer hierarchical relation structure (HRS), i.e., semantically similar relations can make up relation clusters and some relations can be further split into several fine-grained sub-relations. Relation clusters, relations and sub-relations can fit in the top, the middle and the bottom layer of three-layer HRS respectively. To this end, in this paper, we extend existing KGE models TransE, TransH and DistMult, to learn knowledge representations by leveraging the information from the HRS. Particularly, our approach is capable to extend other KGE models. Finally, the experiment results clearly validate the effectiveness of the proposed approach against baselines.

1 Introduction

Knowledge Graphs (KGs) are extremely useful resources for many AI-related applications, such as question answering, information retrieval and query expansion. Indeed, KGs are multi-relational directed graphs composed of entities as nodes and relations as edges. They represent information about real-world entities and relations in the form of knowledge triples, which is denoted as (h, r, t), where h and t correspond to the head and tail entities and r denotes the relation between them, e.g., $(Donald\ Trump, presidentOf, USA)$. Large scale, collaboratively created KGs , such as Freebase (Bollacker et al., 2008), WordNet (Miller, 1994), Yago (Suchanek et al., 2007), Gene Ontology (Sherlock, 2009), NELL (Carlson et al., 2010) and Google's KG[1], have recently become available. However, despite the impressively large sizes, the coverage of most existing KGs are far from complete. This has motivated research in knowledge base completion task, which includes KGE methods aiming to embed entities and relations in KGs into low-dimensional embeddings.

In the literature, there are a number of studies about KGE models. These models embed entities and relations into latent vectors and complete KGs based on these vectors, such as TransE (Bordes et al., 2013), TransH (Wang et al., 2014) and TransR (Lin et al., 2015b). However, most of the existing works simply embed relations into vectors. Less efforts have been made for investigating the rich information from the relation structure. Indeed, in this research, we define a three-layer hierarchical relation structure (HRS), which can be conformed by relation clusters, relations and sub-relations in KGs.

- **Relation clusters**: Semantically similar relations are often observed in Large-scales KGs. For example, the relation *'producerOf'* and *'directorOf'* may be semantically related if both of them describe a relation between a person and a film. These semantically similar relations can make up relation clusters. We believe the information from semantically similar relations is of great value, and relations in the same group can be trained in a collective way to facilitate the knowledge sharing when learning the embeddings of related relations.

- **Relations**: A relation connects the head and

[1] https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3198–3207
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

tail entities in a knowledge triple, denoted as (h, r, t), where h and t correspond to the head and tail entities and r denotes the relation between them.

- **Sub-relations**: There are relations that have multiple semantic meanings and can be split into several sub-relations. For example, the relation $partOf$ has at least two semantics: location-related as $(New\ York$, $partOf$, $USA)$ and composition-related as $(monitor$, $partOf$, $television)$. We believe the sub-relations can give fine-grained descriptions for each relation.

The relation clusters, relations and sub-relations correspond to the top, middle and bottom layer of the three-layer HRS.

In this paper, we extend state-of-the-art models TransE (Bordes et al., 2013), TransH (Wang et al., 2014) and DistMult (Yang et al., 2015) to learn knowledge representations by leveraging the rich information from the HRS. Moreover, the same technique can easily be used to extend other state-of-the-art models and utilize the HRS information. In the proposed models, for each knowledge triple (h, r, t), the embedding of r is the sum of three embedding vectors, which correspond to the three layers of the HRS respectively and therefore, the information from the HRS is leveraged. Particularly, instead of using additional information like text or paths, our model simply use the knowledge triples in KGs and the rich information from the HRS. Extensive experiments on popular benchmark data sets demonstrate the effectiveness of our models.

In summary, we highlight our key contributions as follows,

1. We propose a technique by making use of the HRS information to conduct the KGE task, and extend three state-of-the-art models to utilize this technique. The technique can be easily applied to other KGE models.

2. Our proposed models don't use additional information like text or paths, instead, we only use the knowledge triples in KGs and take advantage of the rich information from the HRS.

3. We evaluate our models on popular benchmark data sets, and the results show that our extended models achieve substantial improvements against the original models as well as other state-of-the-art baselines.

2 Preliminaries and Related Work

We extend three popular KGE models by leveraging the HRS information in this study. Therefore, in this section, we first introduce the three existing models TransE (Bordes et al., 2013), TransH (Wang et al., 2014) and DistMult (Yang et al., 2015) in detail. Then, we further summarize other state-of-the-art models on the topic of KGE.

2.1 TransE, TransH and DistMult

Recently, a number of KGE models have been proposed. These methods learn low-dimensional vector representations for entities and relations (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015b).

TransE (Bordes et al., 2013) is one of the most widely used model, which views relations as translations from a head entity to a tail entity on the same low-dimensional hyperplane, i.e, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds. This indicates that $\mathbf{t}$ should be the nearest neighbor of $\mathbf{h} + \mathbf{r}$. In this case, the score function of TransE is defined as

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_n} , \qquad (1)$$

which can be measured by L_1 or L_2 norm. Positive triples are supposed to have lower scores than negative ones.

TransH (Wang et al., 2014) introduces a mechanism of projecting entities into relation-specific hyperplanes that enables different roles of an entity in different relations. TransH models the relation as a vector $\mathbf{r}$ on a hyperplane $\mathbf{w}_r$ and assumes that $\mathbf{h}_\perp + \mathbf{r} \approx \mathbf{t}_\perp$ when (h, r, t) holds, where $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ are the projection of $\mathbf{h}$ and $\mathbf{t}$ in the relation-specific hyperplane. The score function of TransH is defined as

$$f_r(h, t) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2 , \qquad (2)$$

where $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r$, $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r$ and $\|\mathbf{w}_r\|_2 = 1$. Like triples in TransE, positive triples in TransH should have lower scores than negative ones.

DistMult (Yang et al., 2015) adopts a bilinear score function to compute the scores given (h, r, t) triples. The score function is defined as

$$f_r(h, t) = \mathbf{h}\mathbf{M}_r\mathbf{t}, \qquad (3)$$

where $\mathbf{M}_r$ is a relation-specific diagonal matrix, which represents the characteristics of a relation. Different from TransE and TransH, positive triples should have larger scores than negative ones.

2.2 Other KGE Models

Besides TransE, TransH and DistMult, there are also many models on the topic of KGE. TransR (Lin et al., 2015b) embeds entities and relations into separate entity space and relation-specific spaces. ComplEx (Welbl et al., 2016) extends DistMult to embed entities and relations into complex vectors instead of real-valued ones. HolE (Nickel et al., 2016) employs circular correlations to create compositional representations. ProjE (Shi and Weninger, 2017) adopts a two-layer network to embed entities and relations. Other KGE models also try to embeds entities and relations in various ways, such as Unstructured Model (Bordes et al., 2012a, 2014), Structured Embedding (Bordes et al., 2012b), Single Layer Model (Socher et al., 2013), Semantic Matching Energy (Bordes et al., 2012a, 2014), NTN Model (Socher et al., 2013), etc.

Many efforts have been devoted to building models using additional information like paths or text. For instance, PTransE (Lin et al., 2015a) and R-GCN (Schlichtkrull et al., 2017) use paths as additional information, while DKRL (Xie et al., 2016) and SSP (Xiao et al., 2017) adopt text to assist the embedding task.

Some KGE works focus on making use of the information from relations. CTransR (Lin et al., 2015b), TransD (Ji et al., 2015) and TransG (Xiao et al., 2016) try to find fine-grained representations for each relation. However, these works didn't utilize the information from semantically similar relations and the HRS is also not exploited. Different from the above studies, we believe semantically similar relations can make up relation clusters, and some relations may have multiple semantic meanings and can be split into fine-grained sub-relations. In this paper, we take advantage of the three-layer HRS and conduct the KGE task by extending three widely used models.

3 Methodology

In this section, we provide the technical details of how to extend existing KGE models by leveraging the HRS information. We first formally define the HRS and its integration with existing models.Then we introduce the new loss functions of extended models TransE-HRS, TransH-HRS and DistMult-HRS. Finally, two variants of the HRS models and implementation details are provided.

3.1 Hierarchical Relation Structure

Given a KG $\mathcal{G} = \{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ are the entity (node) set and relation (edge) set respectively. We believe the relations in KGs can make up relation clusters as well as be split into fine-grained sub-relations. On the one hand, large scale KGs always have semantically related relations. The information from semantically similar relations is of great value and these relations should be trained in a collective way. In this way, meaningful associations among related relations can be utilized and less frequent relations can be enriched with more training data. On the other hand, some relations may have multiple semantic meanings and can be split into several sub-relations, which can provide fine-grained descriptions for each relation. In general, relations in KGs conform to a three-layer HRS, as shown in Figure 1. The HRS include a relation cluster layer, a relation layer and a sub-relation layer, which are denoted in yellow, green and blue in Figure 1 respectively.

For a triple (h, r, t) in the HRS model, the embedding of r is comprised of three parts: the relation cluster embedding $\mathbf{r}_c$, relation-specific embedding $\mathbf{r}'$ and sub-relation embedding $\mathbf{r}_s$, which is denotes as

$$\mathbf{r} = \mathbf{r}_c + \mathbf{r}' + \mathbf{r}_s. \tag{4}$$

According to the above equation, the embedding of each relation can leverage the information from the three-layer HRS. The relation clusters and sub-relations are determined by k-means algorithm based on the results of TransE:

- **Relation clusters**. We first run TransE on a given data set and obtain the embeddings of relations $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, ..., $\mathbf{r}_{|\mathcal{R}|}$, where $|\mathcal{R}|$ is the number of relations. Then, the k-means algorithm is applied on these embeddings. In this way, we get relation clusters $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, ..., $\mathcal{C}_{|\mathcal{C}|}$, where $\mathcal{C}$ is the set of relation clusters. Previous studies have shown that the embeddings of semantically similar relations locate near each other in the latent space (Yang et al., 2015). In this way, we are able to find relation clusters composed of semantically related relations.

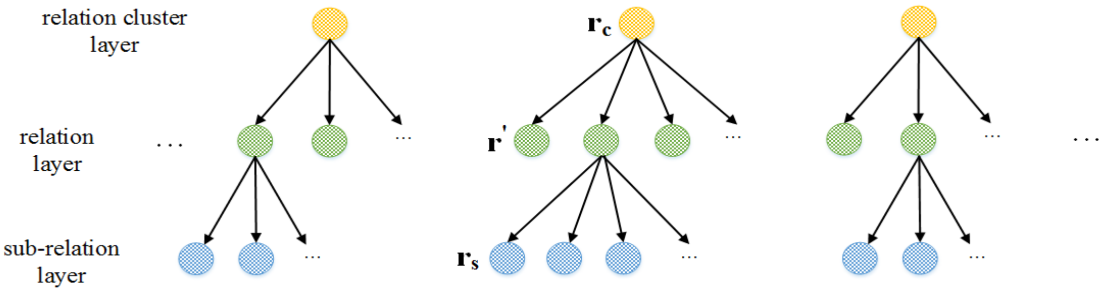


Figure 1: Hierarchical Relation Structure

- **Sub-relations**. TransE assumes that $\mathbf{t} - \mathbf{h} \approx \mathbf{r}$ when (h, r, t) holds. For each triple (h, r, t), we define that $\widehat{\mathbf{r}} = \mathbf{t} - \mathbf{h}$, where $\mathbf{h}$ and $\mathbf{t}$ are obtained from the results of TransE. For each relation, we collect all the $\widehat{\mathbf{r}}$ and adopt the k-means algorithm to cluster these vectors into several groups $\mathcal{S}_1^r$, $\mathcal{S}_2^r$, $\mathcal{S}_3^r$, ..., $\mathcal{S}_{n_r}^r$, where n_r is the number of sub-relations for relation r. Each group corresponds to a fine-grained sub-relation.

3.2 Loss Function

The loss of the extended HRS model is comprised of two parts, as is shown in Equation (5),

$$L_{Total} = L_{Orig} + L_{HRS}, \tag{5}$$

where L_{Orig} is the loss function of the original model, while L_{HRS} is the loss function for the HRS information.

We know that TransE, TransH and DistMult all adopt a margin-based ranking loss. Taking TransE as an example, the loss function of TransE for the first part L_{Orig} is shown as Equation (6),

$$L_{Orig} = \sum_{c=1}^{|\mathcal{C}|} \sum_{r \in \mathcal{C}_c} \sum_{(h,r,t) \in \triangle_r} \sum_{(h',r,t') \in \triangle'_r} [\gamma + f_r(h,t) - f_r(h',t')]_+, \tag{6}$$

where $[x]_+ = max(0, x)$, $\triangle_r$ denotes the set of positive triples for relation r and $\triangle'_r = \{(h', r, t)|h' \in \mathcal{E}\} \cup \{(h, r, t')|t' \in \mathcal{E}\}$ is the set of negative ones for relation r. γ is the margin separating the positive triples from the negative ones. $f_r(h, t)$ is the score function as shown in Equation (7),

$$f_r(h, t) = \left\| \mathbf{h} + \mathbf{r}_c + \mathbf{r}' + \mathbf{r}_s - \mathbf{t} \right\|_{L_n}, \tag{7}$$

which can be measured by L_1 or L_2 norm. Positive triples are supposed to have lower scores than negative ones.

The second part, L_{HRS}, is composed of three regularized terms, which is shown in Equation (8),

$$L_{HRS} = \lambda_1 \sum_{\mathbf{r}_c \in \mathcal{C}} \|\mathbf{r}_c\|_2^2 + \lambda_2 \sum_{\mathbf{r}' \in \mathcal{R}} \left\|\mathbf{r}'\right\|_2^2 + \lambda_3 \sum_{\mathbf{r}_s \in \mathcal{S}} \|\mathbf{r}_s\|_2^2, \tag{8}$$

where $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{|\mathcal{C}|}\}$ is the set of relation clusters, $\mathcal{S} = \{\mathcal{S}_1^r, \mathcal{S}_2^r, \mathcal{S}_3^r, ..., \mathcal{S}_{n_r}^r | r \in \mathcal{R}\}$ is the set of fine-grained sub-relations, n_r is the number of sub-relations for relation r. λ_1, λ_2 and λ_3 are trade-off parameters. Large value of λ_1 will result in the separate training of each relation, while large value of λ_2 will lead to all relations in the same relation cluster sharing the same embedding vector. λ_3 should be larger than λ_1 and λ_2 to restrict $\mathbf{r}_s$ to be a small value, i.e., the sub-relations from the same relation should be close.

3.3 Variants of the HRS Model and Implementation details

Additionally, we introduce two variants of the HRS model: the top-middle model and the middle-bottom model. The top-middle model only uses the HRS by leveraging the information from the top to the middle layer. For this model, the relation embedding and the loss for HRS is defined as Equation (9) and (10).

$$\mathbf{r} = \mathbf{r}_c + \mathbf{r}', \tag{9}$$

$$L_{HRS} = \lambda_1 \sum_{\mathbf{r}_c \in \mathcal{C}} \|\mathbf{r}_c\|_2^2 + \lambda_2 \sum_{\mathbf{r}' \in \mathcal{R}} \left\|\mathbf{r}'\right\|_2^2. \tag{10}$$

While the middle-bottom model only utilizes the information from the middle to the bottom layer. The relation embedding and HRS loss are defined as Equation (11) and (12).

$$\mathbf{r} = \mathbf{r}' + \mathbf{r}_s, \tag{11}$$

$$L_{HRS} = \lambda_2 \sum_{\mathbf{r}' \in \mathcal{R}} \|\mathbf{r}'\|_2^2 + \lambda_3 \sum_{\mathbf{r}_s \in \mathcal{S}} \|\mathbf{r}_s\|_2^2 . \quad (12)$$

The learning process of the extended models is carried out by using the Adam (Kingma and Ba, 2014) optimizer. For the extended models of TransE, all the entity and relation embedding parameters are initialized with a uniform distribution $U\left[-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}}\right]$ following TransE, where k is the dimension of the embedding space. For the extended models of TransH and DistMult, we initialize these parameters with the results of TransE. For the relation cluster embeddings and sub-relation embeddings, we initialize all the parameters with the value of zero.

4 Experiments

4.1 Data Sets

In this research, we evaluate the performances of our extended models on popular benchmarks FB15k (Bordes et al., 2013), FB15k-237 (Toutanova and Chen, 2015), FB13(Socher et al., 2013), WN18 (Bordes et al., 2013) and WN11 (Socher et al., 2013). FB15k, FB15k-237 and FB13 are extracted from Freebase (Bollacker et al., 2008), which provides general facts of the world. WN18 and WN11 are obtained from WordNet (Miller, 1994), which provides semantic knowledge of words. FB15k-237 and WN18 are used for the task of link prediction, FB13 and WN11 are used for the triple classification task, while FB15k is used for both tasks. The statistics of the five data sets are summarized in Table 1.

Table 1: Statistics of the Five Datasets.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#triples in Train/Valid/Test
FB15k	14,951	1,345	483,142 / 50,000 / 59,071
FB15k-237	14,541	237	272,115 / 17,535 / 20,466
FB13	75,043	13	316,232 / 5,908 / 23,733
WN18	40,943	18	141,442 / 5,000 / 5,000
WN11	38,696	11	112,581 / 2,609 / 10,544

4.2 Baselines

To demonstrate the effectiveness of our models, we compare results with the following baselines.

- TransE (Bordes et al., 2013): one of the most widely used KGE models.
- TransH (Wang et al., 2014): a KGE model which adopts relation-specific hyperplanes to lay entities and relations.
- DistMult (Yang et al., 2015): a state of the art model which uses a bilinear score function to compute scores of knowledge triples.
- CTransR (Lin et al., 2015b): a pioneering KGE model which exploits fine-grained sub-relations for each relation.
- TransD (Ji et al., 2015): an improvement of CTransR, which embeds KGs using dynamic mapping matrices.
- TransG (Xiao et al., 2016): the first generative KGE model that uses a non-parametric bayesian model to embed KGs.

4.3 Link Prediction

Link prediction, a.k.a. knowledge graph completion, aims to fill the missing values into incomplete knowledge triples. More formally, the goal of link prediction is to predict either the head entity in a given query $(?, r, t)$ or the tail entity in a given query $(h, r, ?)$.

4.3.1 Experimental Settings

All the parameters are set by some preliminary test. For TransE-HRS, TransE-top-middle and TransE-middle-bottom, λ_1, λ_2, λ_3 and the margin γ are set as $\lambda_1 = 1e-5, \lambda_2 = 1e-4, \lambda_3 = 1e-3, \gamma = 2$. For the extended models of TransH, we set the parameters as $\lambda_1 = 1e-5, \lambda_2 = 1e-5, \lambda_3 = 1e-3, \gamma = 1$. For the extended models of DistMult, the parameters are set as $\lambda_1 = 1e-5, \lambda_2 = 1e-4, \lambda_3 = 1e-3, \gamma = 1$. For all the above models, the learning rate ς, batch size b and embedding size k are set as $\varsigma = 1e-3, b = 4096, k = 100$. The L_1 norm is adopted by the score function of TransE and its extended models. The number of relation clusters are set as 300, 120 and 10 for FB15k, FB15k-237 and WN18 respectively. For all the data sets, we generate 3 sub-relations for relations that have more than 500 occurrences in the training set. For all the extended models and baselines, we produce negative triples following the "bern" sampling strategy which was introduced in TransH (Wang et al., 2014). For baselines TransE, TransH and DistMult, the embedding parameters of entities and relations are initialized the same way as the extended models for a fair comparison.

In the test phase, we replace the head and tail entities with all the entities in KG in turn for each triple in the test set. Then we compute a score for each corrupted triple. Note that for each corrupted triple (h', r, t'), the sub-relation is determined by $\mathbf{t}' - \mathbf{h}'$, i.e., the k-means model is adopted to assign $\mathbf{t}' - \mathbf{h}'$ to a specific sub-relation of r. We rank all the candidate entities according to the scores. Specifically, positive candidates are supposed to precede negative ones. Finally, the rank of the correct entity is stored. We compare our models with baselines using the following metrics: (1) Mean Rank (MR, the mean of all the predicted ranks); (2) Mean Reciprocal Rank (MRR, the mean of all the reciprocals of predicted ranks); (3) Hits@n (Hn, the proportion of ranks not larger than n). Lower values of MR and larger values of MRR and Hn indicate better performance. All the results are reported in the "filtered" setting (Bordes et al., 2013).

4.3.2 Experimental Results

Evaluation results are shown in Table 2. We divide all the results into 4 groups. The second, third and forth group are results of TransE, TransH, DistMult and their extended models respectively, while the first group are results of other state-of-the-art competitors. Results in bold font are the best results in the group and the underlined results denote the best results in the column. From Table 1, we have the following findings: (1) Our extended models outperform the original models, which indicates that the information learned from the HRS is valuable; (2) For WN18, the results from 'top-middle' models of TransE, TransH and DistMult are worse than the original models, and HRS models can't outperform middle-bottom ones. We conjecture the reason lies as follows: WN18 has only 18 relations and the semantic correlation among relations is small. In this case, the information learned from the top to the middle layer of the HRS may lead to worse results since for each relation, even though the information learned from semantically similar relations are useful, the information learned from unrelated relations may damage the results. The results indicate that HRS models are especially useful for KGs with dense semantic distributions over relations; (3) For WN18, TransE-middle-bottom and DistMult-middle-bottom achieve the best results on MRR, Hits@10, Hits@3 and Hits@1 while failing to get the best results on MR in the same group. Further analysis shows that in the results of TransE-middle-bottom, 56 test triples get ranks more than 10000, leading to more than 110 MR loss. While in the results of DistMult-middle-bottom, there exist 37 test triples whose ranks are more than 7000, which would lead to about 50 MR loss. Indeed, MR is sensitive to these high ranks, which lead to worse results on the metric of MR; (4) From all the results, based on the good basic model DistMult, the extended models of DistMult can achieve the best performance compared with other state-of-the-art baselines CTransR, TransD and TransG.

We also provide some case studies on relation clusters and sub-relations. Table 3 shows some relation clusters of FB15k. Cluster 1 to 3 are Olympics-related, basketball-related and software-related relations respectively. From Table 3 we can see that semantically related relations can join the same cluster. Table 4 shows some (head, tail) pairs for the sub-relations of *'/educational_institution/education/degree'*. Sub-relation 1 to 3 are about the degree of Doctor, Master and Bachelor respectively. Table 5 gives some (head, tail) pairs for the sub-relations of *'/music/artist/genre'*. Sub-relation 1 and 2 are about rock music and pop music respectively while sub-cluster 3 is about other kinds of music. From Table 4 and 5, we can see that different sub-relations give fine-grained descriptions for each relation.

4.3.3 Parameter Study

In this section, we study the performance affected by the number of relation clusters $\mathcal{N}_1$ as well as the number of sub-relations for each relation $\mathcal{N}_2$. The results in Figure 2 and 3 clearly show that there exists an optimal value of $\mathcal{N}_1$ and $\mathcal{N}_2$ for each dataset. All three models keep achieving better results as we increase the number of clusters from 0 to the optimal value. Then, after $\mathcal{N}_1$ and $\mathcal{N}_2$ exceed the optimal point, the performance starts falling down. The reason lies as: (1) Smaller value of $\mathcal{N}_1$ leads to large-sized relation clusters. Some unrelated relations may join in the same large-sized cluster and degrade the performance of our models. Larger value of $\mathcal{N}_1$ leads to small-sized relation clusters, thus less information can be leveraged by each relation, leading to the unsatisfying performance; (2) Smaller value of $\mathcal{N}_2$ can't provide sufficient representations for each relation and degrade the performance of our models. Larger value of $\mathcal{N}_2$ may lead to lacking of training data for each sub-relation and also result in the unsatisfying performance.

Table 2: Link prediction results on FB15k, FB15k-237 and WN18. We implement TransE, TransH, DistMult and their extended models by ourselves. The code of CTransR, TransD and TransG are taken from https://github.com/thunlp/TensorFlow-TransX, https://github.com/thunlp/KB2E and https://github.com/BookmanHan/Embedding respectively.

	FB15k					FB15k-237					WN18				
	MR	MRR	H10	H3	H1	MR	MRR	H10	H3	H1	MR	MRR	H10	H3	H1
CTransR	81	0.408	0.740	0.573	0.314	279	0.298	0.469	0.301	0.198	228	0.816	0.923	0.842	0.316
TransD	90	0.658	0.781	0.586	0.324	256	0.286	0.453	0.291	0.179	215	0.823	0.928	0.851	0.336
TransG	101	0.672	0.802	0.591	0.322	309	0.304	0.471	0.298	0.182	466	0.830	0.936	0.876	0.764
TransE	91	0.404	0.688	0.493	0.251	375	0.207	0.377	0.227	0.125	**387**	0.408	0.925	0.725	0.067
TransE-top-middle	61	0.463	0.730	0.556	0.315	286	0.258	0.440	0.286	0.170	609	0.402	0.919	0.710	0.058
TransE-middle-bottom	51	0.493	0.738	0.582	0.355	232	0.310	0.486	0.332	0.202	474	**0.496**	**0.945**	**0.890**	**0.112**
TransE-HRS	**49**	**0.510**	**0.767**	**0.610**	**0.361**	**230**	**0.311**	**0.487**	**0.353**	**0.215**	477	0.490	0.943	0.883	0.106
TransH	63	0.394	0.713	0.519	0.210	311	0.211	0.386	0.224	0.132	388	0.437	0.919	0.832	0.039
TransH-top-middle	65	0.477	0.737	0.561	0.308	275	0.272	0.461	0.291	0.185	411	0.416	0.890	0.813	0.034
TransH-middle-bottom	50	0.469	0.742	0.583	0.343	271	0.269	0.466	0.286	0.191	**283**	**0.491**	**0.942**	**0.880**	**0.113**
TransH-HRS	**47**	**0.509**	**0.783**	**0.639**	**0.390**	**243**	**0.309**	**0.491**	**0.346**	**0.216**	296	0.482	0.940	0.861	0.097
DistMult	95	0.642	0.813	0.726	0.523	251	0.244	0.423	0.261	0.159	261	0.806	0.931	0.904	0.713
DistMult-top-middle	85	0.677	0.830	0.746	0.589	243	0.286	0.461	0.291	0.192	246	0.769	0.903	0.853	0.681
DistMult-middle-bottom	83	0.682	0.828	0.758	0.606	246	0.291	0.475	0.306	0.199	226	**0.912**	**0.947**	**0.913**	**0.879**
DistMult-HRS	**72**	**0.739**	**0.846**	**0.799**	**0.661**	**232**	**0.315**	**0.496**	**0.350**	**0.241**	**206**	0.891	0.932	0.901	0.736

Table 3: Examples of Relation Clusters in FB15k

	relations
1	/olympics/olympic_athlete/medals_won./olympics/olympic_medal_honor/country /olympics/olympic_athlete/country./olympics/olympic_athlete_affiliation/country
2	/sports/sports_team/roster./basketball/basketball_roster_position/player, /basketball/basketball_team/roster./sports/sports_team_roster/player /basketball/basketball_team/roster./basketball/basketball_roster_position/player
3	/computer/software/developer, /computer/operating_system/developer, /cvg/computer_videogame/developer

Table 4: Examples of Sub-relations for Relation '/educational_institution/education/degree' in FB15k

	(head, tail)
1	(Munich Institute of Technology, Doctors of Medicine), (California Institute of Technology, Higher Doctorate), ...
2	(Central Michigan College of Education, M.Sc.), (The University of Pittsburgh, M.Sc.), ...
3	(University of Massachusetts, Amherst, Bachelor's Degree), (New Mexico State College, Bachelor's Degree), ...

Table 5: Examples of Sub-relations for Relation '/music/artist/genre' in FB15k

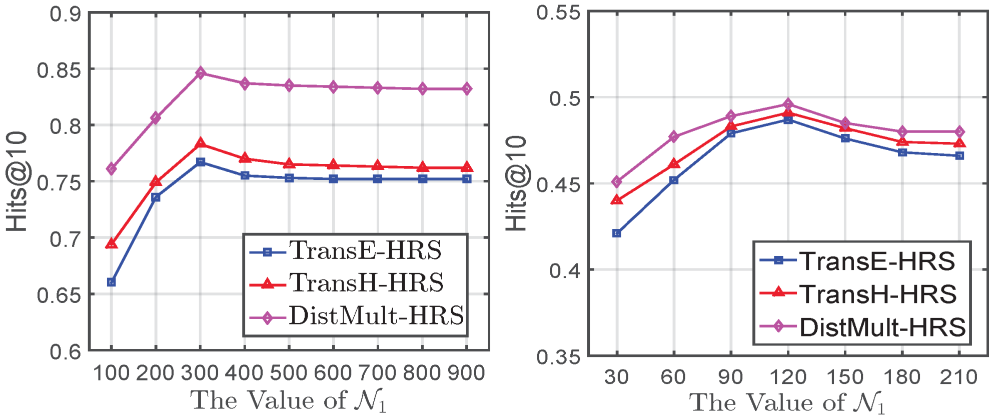
	(head, tail)
1	(Steve Stills, Rock Music), (Velvet Underground, Rock Music), (Benjamin Chase Harper, Rock Music), ...
2	(Justin Beiber, Pop Music), (Natalie Maria Cole, Pop Music), (Peter Thorkelson, Pop Music), ...
3	(Billy Preston, R & B), (Earth Wind Fire, Funk Rap), (Alvin Joiner, Hip-hop), ...

4.4 Triple Classification

In order to testify the discriminative capability of our models, we conduct a triple classification task aiming to predict the label (True or False) of a given triple (h, r, t).

4.4.1 Experimental Settings

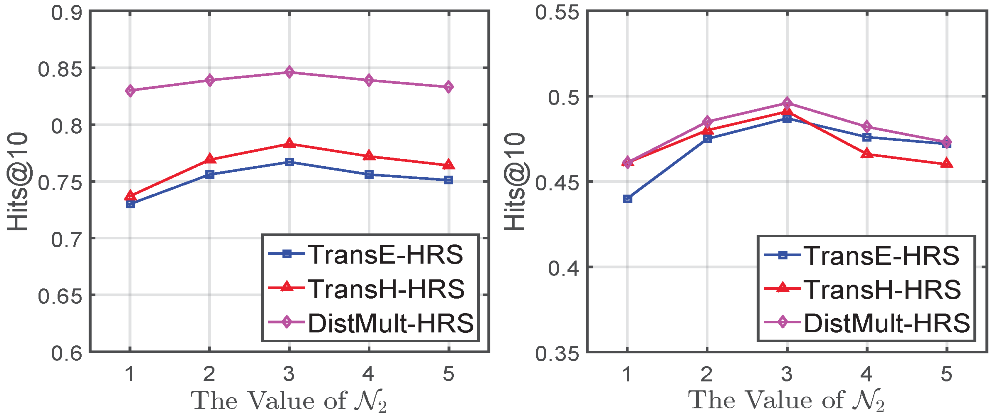
In this paper, we use three datasets WN11, FB13 and FB15k to evaluate our models. The data sets WN11 and FB13 released by NTN (Socher et al., 2013) already have negative triples. The test set of FB15k only contains correct triples, which requires us to construct negative triples. In this study, we construct negative triples following the same setting used for FB13 (Socher et al., 2013). For the extended models of TransE, λ_1, λ_2, λ_3 and γ are set as $\lambda_1 = 1e-5$, $\lambda_2 = 1e-5$, $\lambda_3 = 1e-3$ and $\gamma = 4$. For the extended models of TransH, we set $\lambda_1 = 1e-5$, $\lambda_2 = 1e-4$, $\lambda_3 = 1e-3$ and $\gamma = 5$. While for the extended models of DistMult, parameters are set as $\lambda_1 = 1e-5$, $\lambda_2 = 1e-4$, $\lambda_3 = 1e-2$ and $\gamma = 4$. For WN11 and FB13, we generate 2 sub-relations for each relation. For FB15k, we generate 3 sub-relations for



(a) FB15k

(b) FB15k-237

Figure 2: The Change of Hits@10 with The Value of $\mathcal{N}_1$ Increasing.



(a) FB15k

(b) FB15k-237

Figure 3: The Change of Hits@10 with The Value of $\mathcal{N}_2$ Increasing.

relations that have more than 500 occurrences in the training set. Other parameters are set as introduced in Section 4.3.1. We follow the same decision process as NTN (Socher et al., 2013): for TransE and TransH, a triple is predicted to be positive if $f_r(h,t)$ is below a threshold, while for DistMult, a triple is regarded as a positive one if $f_r(h,t)$ is above a threshold; otherwise negative. The thresholds are determined on the validation set. We adopt accuracy as our evaluation metric.

4.4.2 Experimental Results

Finally, the evaluation results in Table 6 lead to the following findings: (1) Our models outperform other baselines on WN11 and FB15k, and obtain comparable results with baselines on FB13, which validate the effectiveness of our models; (2) The extended models TransE-HRS, TransH-HRS and DistMult-HRS achieve substantial improvements against the original models. On WN11, TransE-HRS outperforms TransE with a margin as large as 10.9%. These improvements indicates the technique of utilizing the HRS information is capable to be extended to different KGE models. Figure 4

Table 6: Triple Classification Results. The results of baselines on WN11 and FB13 are directly taken from the original paper except DistMult. We obtain other results by ourselves.

Model	WN11	FB13	FB15k	Avg
CTransR	85.7	-	84.4	-
TransD	86.4	89.1	88.2	87.9
TransG	87.4	87.3	88.5	87.7
TransE	75.9	81.5	78.7	78.7
TransH	78.8	83.3	81.1	81.1
DistMult	87.1	86.2	86.3	86.5
TransE-HRS	86.8	88.4	87.6	87.6
TransH-HRS	**87.6**	88.9	**88.7**	**88.4**
DistMult-HRS	**88.9**	89.0	**89.1**	**89.0**

shows the classification accuracy of different relations on WN11. We can see that extended models significantly improve the original models in each relation classification task, which again validate the effectiveness of our models.

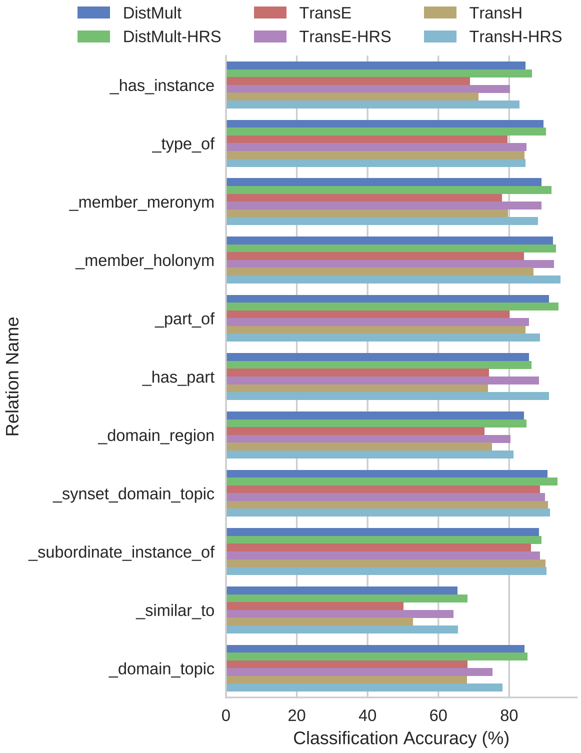


Figure 4: Classification Accuracies of Different Relations on WN11

5 Conclusion

In this paper, we found that relations in KGs conform to a three-layer HRS. This HRS model provides a critical capacity for embedding entities and relations, and along this line we extended three state-of-the-art models to leverage the HRS information. The technique we used can be easily applied to extend other KGE models. Moreover, our proposed models don't need additional information like text or paths, instead, we made full use of the knowledge triples in KGs and the rich information from the HRS. We evaluate our model on the link prediction task and triple classification task. The results show that our extended models achieve substantial improvements against the original models as well as other baseline competitors.

In the future, we will utilize more sophisticated models to leverage the HRS information, e.g, (1) utilize the embeddings of the three layers in a more sophisticated way instead of sum them together; (2) determine the number of relation clusters and sub-relations automatically instead of manually.

Acknowledgements

The research work is supported by the National Key Research and Development Program of China under Grant No. 2018YFB1004300, the National Natural Science Foundation of China under Grant No. 61773361, 61473273, 91546122, Guangdong provincial science and technology plan projects under Grant No. 2015 B010109005, the Project of Youth Innovation Promotion Association CAS under Grant No. 2017146. This work is also partly supported by the funding of WeChat cooperation project. We thank Bo Chen, Leyu Lin, Cheng Niu, Xiaohu Cheng for their constructive advices.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

Antoine Bordes, Xavier Glorot, and Jason Weston. 2012a. Joint learning of words and meaning representations for open-text semantic parsing. In *International Conference on Artificial Intelligence and Statistics*.

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

Antoine Bordes, Nicolas Usunier, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2012b. Learning structured embeddings of knowledge bases. In *AAAI*.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *ACL*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015a. Modeling relation paths for representation learning of knowledge bases. In *EMNLP*.

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

George A. Miller. 1994. Wordnet: a lexical database for english. In *The Workshop on Human Language Technology*.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *AAAI*.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. In *arXiv*.

Gavin Sherlock. 2009. Gene ontology: tool for the unification of biology. *Canadian Institute of Food Science and Technology Journal*, 22(4):415.

Baoxu Shi and Tim Weninger. 2017. Proje: Embedding projection for knowledge graph completion. In *AAAI*.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago:a core of semantic knowledge. In *WWW*.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *The Workshop on Continuous Vector Space MODELS and Their Compositionality*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. Transg : A generative model for knowledge graph embedding. In *ACL*.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2017. Ssp: Semantic space projection for knowledge graph embedding with text descriptions. In *AAAI*.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *AAAI*.

Bishan Yang, Wentau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.