DEIM Forum 2020 H7-3 (day2 p65)

# Translating Embeddings for Knowledge Graph Completion Utilizing Type Correlations

Ran CHENG† and Mizuho IWAIHARA‡

Graduate School of Information, Production and Systems

Waseda University, Kitakyushu, Japan

E-mail: † cr97@akane.waseda.jp, ‡ iwaihara@waseda.jp

**Abstract** Knowledge graph embedding aims to project entities and relations into a continuous low-dimensional space. Typical structure-based embedding methods, such as TransE, are widely used on knowledge graph completion, which concentrate on embeddings with fact triples indicating relations between entities. In fact, such methods only use structural features to learn embeddings, but most knowledge graphs provide specific taxonomies for entities, which have not been well utilized by existing methods. In this paper, we propose a novel knowledge graph embedding method, which is able to take advantages of both fact triples and type information for entities. We assume that if two entities are correlated according to their belonging types, embeddings should be closer to each other in the low-dimensional space. In other words, when the categories of two entities are similar, they should hold similar relations. Based on this assumption, the embedding of an entity is responsible for both modeling the corresponding fact triples and modeling its type information in this model. Type information is encoded into a label for each entity, producing type-based label embeddings, which capture type correlations between entities. In this way, conventional embedding methods, which solely learn embeddings from fact triples, can be improved by reflecting type correlations between entities.

**Keyword** Knowledge graph completion, Translating embeddings, Type correlations

## 1. Introduction

Knowledge graphs (KG)[1] are directed graphs which provide sufficient structured knowledge information, consisting of entities as nodes and relations as edges. Typically, a KG represents knowledge as triplets i.e., (*head entity, relation, tail entity*), indicating there exists a relation between two entities. KGs encode structured information of billions of fact triples, however, due to the quantities of entities and relations, it is still far from completeness. Therefore, knowledge graph completion (KGC) is derived to predict new triples according to the existing fact triples in KGs.

Recently, translation-based methods [14] are becoming increasingly popular. They embed each object in a KG into a continuous low-dimension vector space and view fact triples as translations in the embedding space. TransE [4] is a conventional translation-based method, it holds the view that translations are natural transformations for representing facts in the KG. For a triplet $(h,r,t)$, the embedding of the tail entity $t$ should be close to the embedding of the head entity $h$ plus some vector that depends on the relationship $r$. However, such methods only learn embeddings from topological structure, where type information or schema of each entity, supported in most of KGs, is not utilized. In Fig. 1 we show the belonging types of entity *David Fincher* which is sampled from Freebase[2], a significant KG maintained by Google. For instance, we can extract a triple *(Obama, eat, pear)* from the example "*Obama eats pears*". Since *Obama* belongs to type *people* and *pear* belongs to type *food*, other entities like *David* and *banana* might also also form a triple which is (*David, eat, banana*). This is consistent with human cognition.

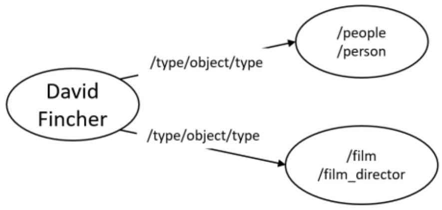


**Figure 1. Freebase triples containing type information:** Two triples linking a *subject* (i.e., David Fincher) through *predicates* (i.e., /type/object/type) to *objects* (i.e., /people/person, /film/film_director).

We believe that such type information can augment KGC task during the embedding process, because it introduces additional information into entity representations. To utilize both kinds of information, we propose a model named TransType to learn two representations for entities. Structure-based representations capture information in

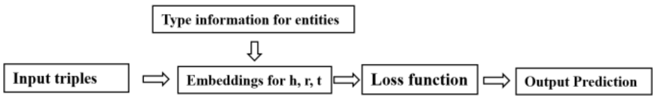


**Figure 2. Framework of our model**

fact triples of KGs, while type-based representations capture inner connections among entities. Specifically, type information is embedded as an added feature for corresponding entities. Following the idea of translating embeddings, we let the loss function include collected type information. Finally, structure-based and type-based information jointly formulate the final embeddings. Fig. 2 shows the framework of our model. Type information is added into the embedding process to calculate the final loss function.

The rest of the paper will be organized in two parts. Section 2 surveys related work. In Section 3 we show construction of type correlation graphs. Then we show the model to combine with translating embedding models.

## 2. Related work

Wide varieties of embedding methods for KGs have been proposed in recent years. The basic idea is to model multi-relational data by embedding both the entities and relations into a low dimension space. Bordes first proposed TransE [4] by embedding relations as the translation operation between the entities, which performs good in link prediction on very large databases, due to its low-dimensional embedding spaces and reduced parameters.

Wang et al. [11] pointed out that TransE embeds entities with unique representations, which could cause problems when it comes to relations other than one-to-one. Based on this, they proposed TransH by projecting entity embeddings into specific hyperplane according to different relations involved. Both [4] and [11] embed entities and relations in the same space.

Assuming that different relations might focus on distinct aspects of the entity, TransR [8] embeds entities and relations in different spaces and use a transfer matrix for mapping. TranSparse [5] replaces the transfer matrix with adaptive sparse matrix based on the number of relation-linked entities to deal with the heterogeneity and imbalance problems. It also considers the diversity of entities and reduces the parameters by using vector-only multiplication instead of matrix-included one for relation-specific entity embedding.

Following the ideas of separating embedding space for entities and relations, TransAt [9] tried to realize attention mechanism in relation-related transformation. Two-stage learning is proposed, such that first collecting candidate entities from correct categories and then fine-graining the difference by attention. However, TransAt uses clustering to generate categories instead of the ground truth ones.

For introducing more information into entity representations, DKRL [13] first encodes entity descriptions by continuous bag-of-words and deep convolutional neural models and combines with TransE.

Based on our observation, all of the former ideas try to increase the complexity of embedding spaces for both entities and relations over the original TranE method, in order to give a better representation of the multi-relational data. Inspired by category filtering of TransAt and entity description encoding of DKRL, our idea is to include the ground truth type information into translating embedding models, which have not been utilized in knowledge graph completion. Assuming that entities with similar types are likely to have similar relations, we propose an additional model to embed type correlations into the score function. Using the same idea of translating embedding, all of the types according to the specific entity are represented as added features to better separate the embedding space for related entities.

## 3. Embedding model with type correlations

In this section, we describe how we generate a type correlation graph according to type information in KGs, and explain our proposing model.

To utilize both structure and type information among entities, we propose two types of embeddings for entities, namely *translating embeddings* and *type-based embeddings*. Translating embeddings are suitable for capturing topological information in fact triples of KGs, while type-based representations are suitable for capturing category information in entity taxonomies. These two embeddings jointly produce representations for entities.

### 3.1 Problem formulation

We first introduce the notations used in this paper. A knowledge graph is defined as $G = (E, R, T)$, where $E$ is an entity set, $R$ is a relation set and $T$ is a triple set. Given a triple $(h, r, t)\epsilon T$, where $h, t\epsilon E$ stand for head and tail entities, respectively, and $r\epsilon R$ stands for relations. $T', \; where \; T' \subset E \times R \times E \; and \; T' \cap T = \emptyset$, denotes missing but valid triples in the triple set, and $\tilde{T}$ denotes triples

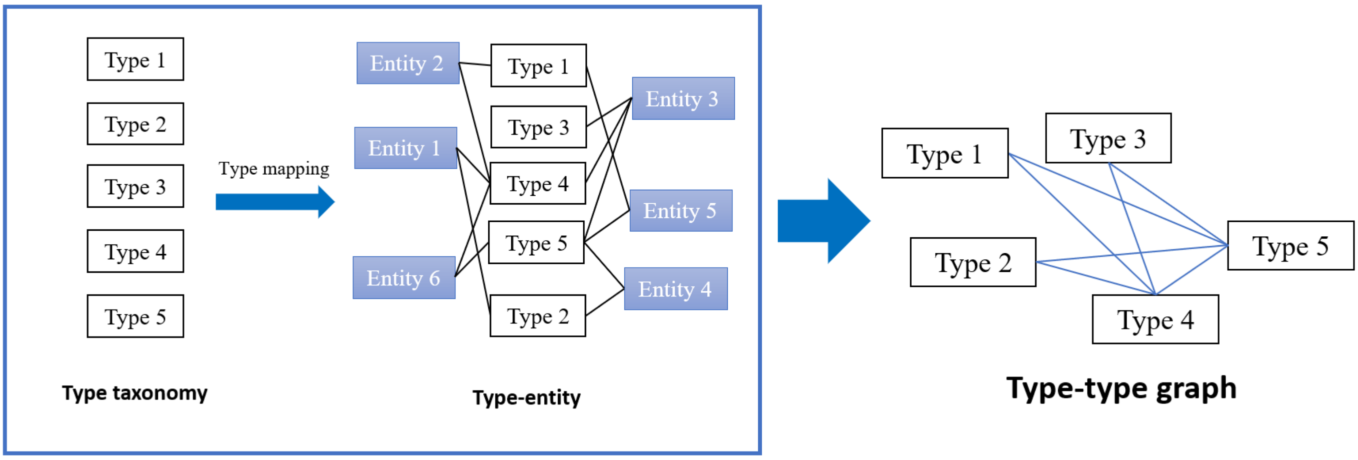


**Figure 3. Type correlation graph construction**

which are randomly selected from $T$.

**Translating embeddings:** $h$ and $t$ are translating embeddings for head and tail entities. This kind of embeddings is the same as those learned from existing translation-based models such as TransE.

**Type-based embeddings:** $h_{type}$ and $t_{type}$ are type-based embeddings for head and tail entities which are built from entity taxonomies. We calculate type-based correlation score $c_{ht}$ and supply it into embedding process to construct type-based embeddings.

### 3.2 Type correlation graph construction

To capture the inner category correlation between types, we extract type information from the knowledge base or feature type hierarchy, and construct a type correlation graph to help evaluate type correlation between entities, and produce type-based correlation vectors. In this graph, each node represents a type name, and the weight of each link represents the correlation score between two nodes.

The process of type correlation graph construction is described in Figure 3. Feature type taxonomy lists a set of type name which is extracted from FIGER dataset [7], which has one-to-many mappings with Freebase taxonomy. Type-entity facts are extracted from Freebase (triplets with "type.instance" as predicate). In the type correlation graph, the blue line indicates the weight between two nodes(types), indicating correlation score between two corresponding types.. Firstly, we map feature type taxonomy to type-entity facts, and generate feature type-entity facts. Then we generate the type correlation graph according to shared entities in the feature type-entity facts.

We utilize entity-type facts in the knowledge base to measure type correlation. According to Ren et al. [10], the correlation score between types is proportional to the number of entities they share in the knowledge base. The correlation score is defined as follows:

$$s_{kk'} = (|\varepsilon_k \cap \varepsilon_{k'}|/|\varepsilon_k| + |\varepsilon_k|/|\varepsilon_{k'}|)/2 \qquad (1)$$

$$\varepsilon_k = \{e \mid (e, k) \in \varphi_{KB}\} \qquad (2)$$

where $\varepsilon_k$ denotes the set of entities assigned with type $k$ in $\varphi_{KB}$, and $|\varepsilon_k|$ denotes the cardinality of set $\varepsilon_k$.

Figure 4 shows the visualization of type correlation graph. The source of feature type taxonomy is extracted from FIGER dataset[7], and the type-entity facts are extracted from Freebase. Each node represents a type, and the distance between two nodes is proportional to their corresponding correlation score. Also, the size of each node is proportional to the number of correlating types it has.

With the help of a type correlation graph, we can measure the inner connection between entities in taxonomies, and thus obtain entity type embedding.

### 3.3 Methodology

In this section, we present our proposing model. We assume that if two entities are correlating according to their belonging types, embeddings should be closer to each other in the low-dimensional space. To implement this idea, we first obtain type-based correlation vectors based on the type correlation graph, then propose a general approach utilizing correlation vectors to do entity

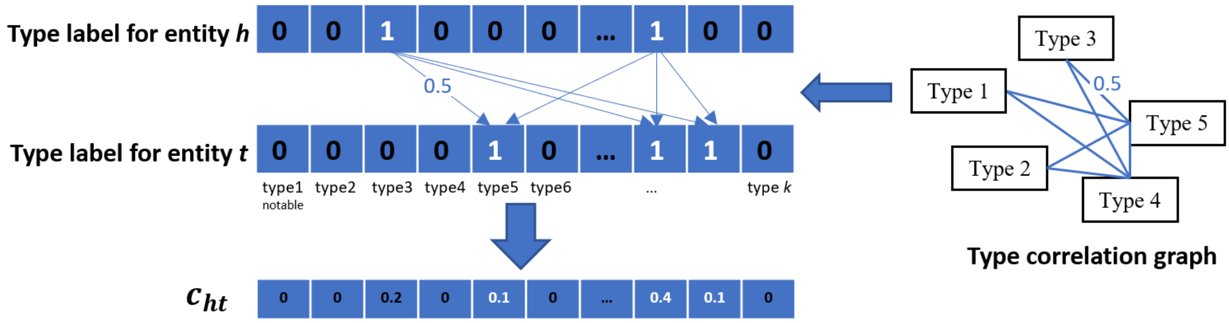


**Figure 5. Type-based correlation vector $c_{ht}$**

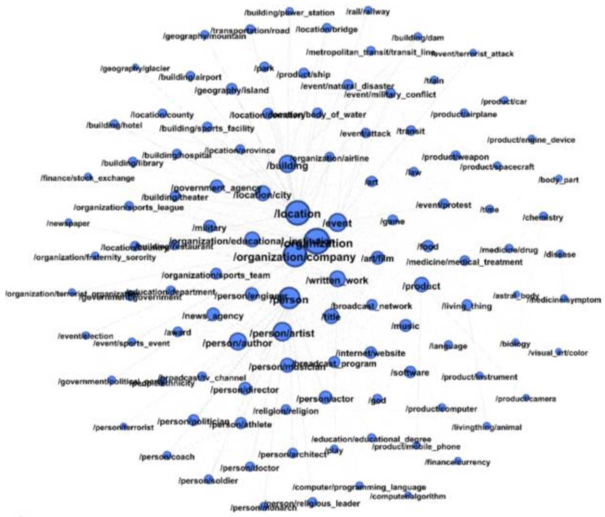

**Figure 4. Visualization of type correlation graph**

type embedding, and finally modify the margin-based loss function in TransE, to let its margin parameter include type information.

**Type-based correlation vectors.** We first define $\boldsymbol{c}_{ht}$ as the type-based correlation vector between entity $h$ and $t$:

$$\boldsymbol{c}_{ht} = \alpha \cdot c_{notable} + (1-\alpha)\, c_{other}, \qquad (3)$$

where $c_{notable}$ denotes the correlation score between notable types which are obtained from Freebase taxonomy (triplets with "*notable_for*" as predicate) [ ]. In Freebase, certain types are notable because they hold a large number of entities, or they might link to many other types. FB defines that each entity has only one notable type. $c_{other}$ denotes the correlation score between types which are obtained from the feature type set, $0 < \alpha < 1$ is a parameter that balances the importance of these two kinds of types.

Figure 5 shows the process of generating type-based correlation vector $\boldsymbol{c}_{ht}$. Each element in the type labels indicates whether or not this entity belongs to the corresponding type. The first element (type1) in the label denotes the notable type. Firstly we generate type labels for entities according to feature type-entity facts. Each label is a vector with 100 elements, describing each entity's belonging types. To obtain type-based correlation vector between entity $h$ and $t$, we refer to type correlation graph which is described in Section 3.2, using the type correlation score between two vectors. Types have correlation scores with each other according to the type correlation graph. In the final correlation vector, each element is the average of the corresponding type correlation scores.

**Entity type embedding.** After obtaining type-based correlation vectors $\boldsymbol{c}_{ht}$, we formulate entity type embeddings with the following loss function:

$$L_{type} = \sum_{(h,t)\in T} \sum_{(h\prime,t\prime)\in N} \max\left(\left(\gamma + d\left(h_{type} + \boldsymbol{c}_{ht}, t_{type}\right) - d\left(h'_{type} + \boldsymbol{c}_{ht'}, t'_{type}\right)\right), 0\right) \qquad (4)$$

where $d(\cdot)$ denotes dissimilarity function, which can be either the L1 or L2 distance, $\gamma$ is a margin parameter, and $N$ is the set of non-relevant entity pairs such that there is no relation between the two entities. $N$ is constructed as follows: For each triple in the training triple set, replace either the head or tail entity by a random non-relevant entity.

In the loss function of (4), we follow the idea of structure-based embedding in TransE, and add type information to the embeddings process. Instead of only relying on topological structure, we introduce additional information into KGC. We view ($h_{type}$, $c_{ht}$, $t_{type}$) as triplets, and the loss function (4) favors lower values for the training set than for non-relevant entity pair set, and thus obtain the final entity type embedding to augment the final loss function.

**Type-based margin loss function.** In TransE, if triple $(h,r,t)$ holds, then the embedding of $t$ should be close to $h$ plus vector $r$. The loss function favors lower values of the energy for training triplets $(h,r,t)$ than for negatively sampled triplets $(h',r,t')$. Similar to TransE, if triple $(h,r,t)$ holds, then the type embedding of $t$ should be close to $h$ plus vector $r$. To utilize both structure and type information, we propose a type-based margin loss function:

$$L = \sum_{(h,r,t)\in T} \sum_{(\tilde{h},r,\tilde{t})\in T} \max\left(\gamma_{type} + d(h+r,t) - d(\tilde{h}+r,\tilde{t}),\ 0\right) \quad (5)$$

where $(\tilde{h},r,\tilde{t})$ is either the head or tail is replaced by a random entity, and

$$\gamma_{type} = d\left(t_{type},\ h_{type}\right) - d\left(\tilde{t}_{type},\ \tilde{h}_{type}\right) \quad (6)$$

**Algorithm 1:** Learning model

**Input**: Training set S = $\{(h,r,t)\}$, entities and rel. sets E and R, embeddings dimension K.

**Output:** Embeddings for entities and relations

```
initialize r ←uniform(−6/√k, 6/√k) for each relation r ∈ R
        r ← r/||r||  for each r ∈ R
        e ← uniform (−6/√k, 6/√k) for each entity e ∈ E
loop
    e ← e/||e||  for each entity e ∈ E
    S_batch ←sample(S, b) //sample a minibatch of size b
    T_batch ← ∅ //initialize the set of pairs of triplets
    for (h,r,t) ∈ S_batch do
        (h̃, r, t̃)←sample(S_(h,r,t))
    T_batch ← T_batch ∪ {(h,r,t),(h̃, r, t̃)}
    end for
    Update embeddings w.r.t.
        ∑ ∇ (γ_type + d(h + r, t) − d(h̃ + r, t̃))
end loop
```

$\gamma_{type}$ can be considered as the difference of type dissimilarity between pairs $(h,r,t)$ and $(\tilde{h},r,\tilde{t})$. For example, for triplets $(h,r,t)$ and $(h,r,\tilde{t})$, if entities $t$ and $\tilde{t}$ share similar types, $\gamma_{type}$ will become smaller, so $t$ and $\tilde{t}$ will both embed closer to $h$ and $r$ according to the loss function. To avoid overfitting, changes of $\gamma_{type}$ are limited to a small scale, and let translating embeddings make the major influence. Instead of the predefined hyper-parameter $\gamma$, the type-based margin parameter $\gamma_{type}$ adopts to different circumstances and makes embeddings more precise. The detailed learning procedure is described in Algorithm 1.

## 4. Experiment

### 4.1 Datasets

We adopt FB15k [4], a public knowledge graph dataset extracted from Freebase [2] corpus, to evaluate our proposed model on knowledge graph completion. Freebase is a large-scale online collection of structured data harvested from varieties of sources, aiming to allow people and machines to access common information more effectively. Table 1 lists the statistics of FB15k dataset.

**Table 1. Statistics of FB15k**

| Dataset | \|Rel\| | \|Ent\| | #Train | #Valid | #Test |
|---|---|---|---|---|---|
| FB15k | 1,345 | 14,951 | 483,142 | 50,000 | 59,071 |

**Table 2. Statics of feature type-entity facts for FB15k**

| | | | | |
|---|---|---|---|---|
| \|Type\| | 1 | 2 | 3 | 4 |
| \|Entity\| | 1,683 | 3,945 | 4,359 | 2,071 |
| \|Type\| | 5 | 6 | 7 | 8 |
| \|Entity\| | 1,844 | 985 | 62 | 2 |

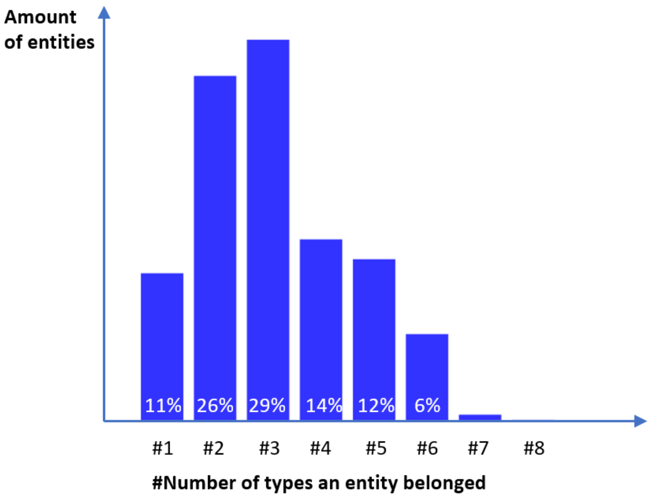


**Figure 6. Histogram of feature type-entity facts**

**Table 3. Experimental results of Knowledge graph completion**

| Metric | Mean Rank | | Hits@10 | |
|---|---|---|---|---|
| | Raw | Filter | Raw | Filter |
| TransE | 243 | 125 | 34.9 | 47.1 |
| TransH | 212 | 87 | 45.7 | 64.4 |
| TransR | 226 | **77** | 48.2 | **68.7** |
| DKRL(CNN)+TransE | **181** | 91 | 49.6 | 67.4 |
| TransType | 190 | 100 | **50.0** | 66.9 |

For feature type extraction, we use the FIGER [7] dataset, which contains 128 entity types and 2-level hierarchy. In FIGER, some types are defined too general, and have correlation with almost all the other types, and thus we view such types as noisy ones. To ensure the efficiency of our type embedding, as well as in accordance with entity embeddings, we eliminate such noisy types in FIGER and extract 100 types as our feature type set. Table 4 shows our cleaned feature type set, where each bold-tag is a rough summary of each box. The box at the bottom right corner contains mixed tags that are hard to be categorized. Utilizing FIGER, we obtain feature entity-type facts for FB15k. We show the statistics of feature type-entity facts of FB15k in Table 2 and Figure 6. The odd-numbered lines and even-numbered lines, respectively, indicate the number of belonging types and the amount of entities, respectively. Figure 6 is a histogram corresponding with Table 2. Over half of the entities have two or three types, and the majority of entities has less than six types.

### 4.2 Settings

For both entity embeddings and entity type embeddings, we use the same parameter settings. The embedding size $n$ is 100, the margin size $\gamma$ is 1, the learning rate is 0.001, and the batch size is 100. We adopt *tanh* as activation function, and L2 distance as dissimilarity function. Training epoch is 1000.

During learning, embeddings for entities and relations are initialized following the random procedure proposed by TransE. In each epoch, a small set of triples is sampled from the training set. Since sampling matters during learning [6][12], we use sampling method called "bern"[11] to sample batches of data and SGD [3] to optimize our loss function.

### 4.3 Knowledge graph completion

The task of knowledge graph completion is to find the missing $h$ or $t$ for a correct triplet $(h, r, t)$. It emphasizes the rank of right object instead of obtaining the best one object.

**Evaluation protocol.** The evaluation of knowledge graph completion contains two metrics: (1) The average rank of correct entities (Mean Rank) (2) The proportion of correct entities ranked in top 10 (Hits@10).

In Mean Rank, the rank of the correct entity is recorded. While for Hits@10, if the correct entity is ranked in top 10, a hit is counted. We split each evaluation into two parts: head entity completion task and tail entity completion task. The final result is calculated by the average result of two tasks.

Although we want to predict new triples in KGs, some triples existing in training and validation set might become invalid in the test set. Such triples are supposed to be the ground truth in the test set rather than being newly predicted. To avoid this issue, from the list of invalid triples we remove all the triples which appear either in the training, valid or test set, so as to ensure all the newly predicted triples are not in these three sets. This evaluation setting is denoted as "Filter" and the original setting is denoted as "Raw."

**Results and discussion.** We show our experimental results in Table 3. For baseline models, TransE is a conventional translation-based method which raises the view that translations are natural transformations for representing facts in the KG, adopting energy function $E(h, r, t) = ||h + r - t||$ to score each triple. TransH [11] and TransR [8] extend TransE in the way that TransH projects entity embeddings into specific hyperplanes according to different relations involved; TransR embeds entities and relations in different spaces and use a transfer matrix for mapping. DKRL [13] introduces neural networks into translation-based methods, reflecting entity descriptions into embeddings by using a CNN to encode text information.

| | | | |
|---|---|---|---|
| **Person** | /person/director | **Organization** | /organization/terrorist_organization |
| /person | /person/coach | /organization | /government/political_party |
| /person/athlete | /person/architect | /organization/sports_team | /government/government |
| /person/author | /person/engineer | /organization/educational_institution | /government_agency |
| /person/actor | /person/soldier | /organization/company | /religion/religion |
| /person/artist | /person/religious_leader | /organization/airline | /news_agency |
| /person/politician | /person/doctor | /organization/fraternity_sorority | /education/department |
| /people/ethnicity | | /organization/sports_league | |
| **Location** | /location/cemetery | **Product** | /product/spacecraft |
| /location | /location/body_of_water | /product | /product/computer |
| /transportation/road | /park | /product/instrument | /game |
| /location/country | /geography/island | /product/car | /food |
| /location/city | /geography/glacier | /product/ship | /internet/website |
| /location/province | /rail/railway | /product/airplane | /train |
| /location/county | | /product/weapon | /software |
| | | /product/engine_device | /newspaper |
| | | /product/camera | |
| **Building** | **Art** | /living_thing | /chemistry |
| /building | /art | /livingthing/animal | /broadcast/tv_channel |
| /building/power_station | /art/film | /broadcast_program | /law |
| /building/restaurant | /visual_art/color | /title | /biology |
| /building/airport | /play | /language | /transit |
| /building/sports_facility | /written_work | /education/educational_degree | /time |
| /building/library | **Event** | /broadcast_network | /finance/stock_exchange |
| /building/hospital | /event | /finance/currency | /god |
| /building/theater | /event/sports_event | /body_part | /computer/programming_language |
| /building/hotel | /event/natural_disaster | /disease | /computer/algorithm |
| /building/dam | /event/attack | /astral_body | |
| | /event/election | | |
| | /event/protest | | |

**Table 4. Feature type set:** It contains a number of 100 types, we separate them into 7 main categories(i.e., person, organization, location, building, art, event, product). Types that are hard to be categorized are located at the bottom right corner.

Our method has an overall improvement over TransE, especially for Hits@10, but does not yield all the other methods. It implies that type information, which has been successfully added into embedding process, could provide a good supplement for translation-based method. TransType may not have a large improvement over other translation-based methods, because structured-based representations can already handle this task. However, instead of solely relying on topological features among entities and relations, our TransType introduces entity taxonomies into embeddings to help entity representations contain more aspects of information. Especially in the situation that certain entities are newly added to the KGs, existing models based on translation-based methods cannot predict triples for such entities, because current triples in the KGs do not contain new entities. Lacking topological information would limit translation-based models to form representations. Nevertheless, TransType could handle this situation because our model refers entity taxonomies to the embedding process, representations for new unknown entities can be generated if their type information is available.

## 5. Conclusion and future work

In this paper, we propose the TransType model for knowledge graph completion. The idea of conventional translation-based embedding methods for knowledge graph completion is to embed each object in a KG into a continuous low-dimension vector space and view fact triples as translations in the embedding space. We find that previous methods only learn representations from topological features in the low-dimension space, ignoring inner type correlation between entities. Therefore we propose a model named TransType, which utilizes type information in translating embeddings for knowledge graph completion. We use structured-based and type-based embeddings to jointly generate embeddings for entities. Structured-based embeddings follow the idea of conventional translation-based model TransE, while type-based embeddings are obtained according to type correlation between entities. Our experimental results prove that TransType indeed has a reinforcement for translation-based methods, and prove its capability of generating embeddings from type information.

In the future, we will explore the following aspects: (1) Our feature type set contains only 100 types, while the Freebase taxonomy contains thousands of types. If we could filter the original taxonomies properly, richer type information would be added to entity representations. (2) Type hierarchy provides inner connection between types, thus we can dig into this hierarchical structure to further exploit type correlation. (3) We prove the effectiveness of applying type information into entity embeddings only with TransE. It is possible to extend our model to other translation-based models such as TransH and TransR. We will investigate more sophisticated models for the above purpose in the future.

## References


[1] R. R. Bakker, 1987. Knowledge Graphs: Representation and Structuring of Scientific Knowledge, Ph.D. thesis, University of Twente, Enschede, ISBN 9001963-4

[2] K. Bollacker; C. Evans; P. Paritosh; T. Sturge; and J. Taylor, 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of KDD, 1247–1250.

[3] L´eon Bottou. Large-scale machine learning with stochastic gradient descent. In COMPSTAT, pages 177–186. Springer, 2010.

[4] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data," in NIPS, 2013, pp. 2787–2795.

[5] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge Graph Completion with Adaptive Sparse Transfer Matrix," in AAAI, 2016, pp. 985–991.

[6] Vibhor Kanojia, Hideyuki Maeda, Riku Togashi, and Sumio Fujita. Enhancing knowledge graph embedding with probabilistic negative sampling. In WWW, pages 801–802. International World Wide Web Conferences Steering Committee, 2017.

[7] X. Ling and D. S. Weld. Fine-grained entity recognition. In AAAI, 2012.

[8] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in AAAI, 2015, pages 2181–2187.

[9] W. Qian, C. Fu, Y. Zhu, D. Cai, and X. He, "Translating Embeddings for Knowledge Graph Completion with Relation Attention Mechanism," in IJCAI, 2018, pp. 4286–4292.

[10] X. Ren, W. He, M. Qu, C. R. Voss, H. Ji, and J. Han, "Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding.," CoRR, vol. abs/1602.05307, 2016.

[11] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph Embedding by Translating on Hyperplanes," in AAAI, 2014, pp. 1112–1119.

[12] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Kr¨ahenb¨uhl. Sampling matters in deep embedding learning. arXiv, 2017.

[13] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation Learning of Knowledge Graphs with Entity Descriptions," in AAAI, pp. 2659-2665, 2016.

[14] Y. Zhua, Z. Guan, S. Tanc, H. Liud, D. Caia, and X. Hea, Heterogeneous hypergraph embedding for document recommendation, 2016. A.Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen, "Knowledge graph embedding by translating on hyperplanes," in AAAI, pages 1112–1119, 2014.