Text-Enhanced Representation Learning for Knowledge Graph

Zhigang Wang[†] **and Juanzi Li**[‡]
Tsinghua University, Beijing, CHINA
[†] wangzg14@mails.tsinghua.edu.cn
[‡] lijuanzi@tsinghua.edu.cn

Abstract

Learning the representations of a knowledge graph has attracted significant research interest in the field of intelligent Web. By regarding each relation as one translation from head entity to tail entity, translation-based methods including TransE, TransH and TransR are simple, effective and achieving the state-of-the-art performance. However, they still suffer the following issues: (i) low performance when modeling 1-to-N, N-to-1 and N-to-N relations. (ii) limited performance due to the structure sparseness of the knowledge graph. In this paper, we propose a novel knowledge graph representation learning method by taking advantage of the rich context information in a text corpus. The rich textual context information is incorporated to expand the semantic structure of the knowledge graph and each relation is enabled to own different representations for different head and tail entities to better handle 1-to-N, N-to-1 and N-to-N relations. Experiments on multiple benchmark datasets show that our proposed method successfully addresses the above issues and significantly outperforms the state-of-the-art methods.

1 Introduction

Knowledge graphs aim at semantically representing the world's truth in the form of machine-readable graphs composed of subject-property-object triple facts. Taking the (h, r, t) (short for (*head entity*, *relation*, *tail entity*)) triples as input, representation learning for knowledge graph represents each entity h (or t) as one low-dimensional vector **h** (or **t**) by defining relation-dependent scoring function $f_r(h, t)$ to measure the correctness of the triple in the embedding space. The learned representations make the knowledge graph essentially computable, and have been proved to be helpful for knowledge graph completion, documentation classification and information extraction [Socher *et al.*, 2013; Bordes *et al.*, 2013; Wang *et al.*, 2014b; Lin *et al.*, 2015b].

Among current knowledge graph representation learning methods, the translation-based methods have achieved the state-of-the-art performance, by regarding each relation as one *translation* from head entity to tail entity. Inspired by [Mikolov *et al.*, 2013], TransE [Bordes *et al.*, 2013] learns the entity and relation embeddings to satisfy $\mathbf{r} \approx \mathbf{t} - \mathbf{h}$ when (h, r, t) holds. TransE is simple, efficient and effective, but has issues when modeling 1-to-N, N-to-1 and N-to-N relations. To address the issue, TransH [Wang *et al.*, 2014b] and TransR [Lin *et al.*, 2015b] are proposed to enable an entity to have different representations for different relations by preliminarily generating the relation-specific entity embeddings with mathematical transformations, which are hyperplane projection for TransH and space projection for TransR. Better results are reported in [Wang *et al.*, 2014b; Lin *et al.*, 2015b]. However, the performance when predicting the entity where multiple entities could be correct is still unsatisfactory, with the average Hits@10 to be about 50% [Lin *et al.*, 2015b].

On the other hand, by learning the embeddings directly from the graph structure, the performance is limited by the structure sparseness of the knowledge graph, which is quite common especially in the domain-specific and non-English situations [Wang *et al.*, 2013]. As we will present in detail in Section 4.2, the performance of TransE is highly influenced by the density of the knowledge graph, with the mean rank of link prediction task to be 102.7 for FB3K, 81.9 for FB6K and 79.5 for FB9K on the same testing dataset.

In order to solve the above problems, which are **low performance on 1-to-N, N-to-1 and N-to-N relations** and **structure sparseness of knowledge graph**, we propose a novel knowledge graph representation learning method by taking advantage of the rich context information in a text corpus. Inspired by the idea of distant supervision [Mintz *et al.*, 2009], we find that the textual context information of entities is helpful to model the semantic relationships in the knowledge graph. As shown in Figure 1, the textual contexts (sets of words here) reveal that "*Avatar*" should be a film and "*James Cameron*" should be a director. And the common set of entity contexts indicates the relationship between the entities should be "*direct*".

In this paper, we propose a novel **t**ext-**e**nhanced **k**nowledge **e**mbedding (**TEKE**) method for knowledge graph representation learning. Given the knowledge graph to be represented and a text corpus, we firstly semantically annotate the entities in the corpus and construct a co-occurrence network composed of entities and words to bridge the knowledge graph and text information together. Based on the co-occurrence

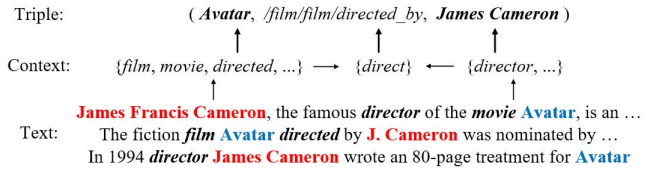

Figure 1: Simple Illustration of Text-enhanced Method.

network we define the textual contexts for entities and relations, and incorporate the contexts to the knowledge graph structure. Finally a normal translation-based optimization procedure is utilized to learn the embeddings of the entities and relations. The key points of the proposed method are as follows:

- We represent each relation's textual contexts in (h, r, t) as the common of h and t's contexts, which enables each relation to own different representations for different head and tail entities to better handle 1-to-N, N-to-1 and N-to-N relations.
- We incorporate the textual contexts to each entity and relation, which greatly expands the semantic structure of the knowledge graph.

Different TEKE methods based on the optimization targets of TransE, TransH and TransR are implemented and extensive experiments have been conducted on link prediction and triple classification with benchmark datasets including WordNet and Freebase. Experiments show that our method can effectively deal with the problems of low performance on 1-to-N, N-to-1 and N-to-N relations and structure sparseness of knowledge graph. In summary, the contributions of this paper are as follows:

1. We propose a novel text-enhanced knowledge embedding method. The incorporation of textual contexts greatly expands the graph structure and successfully handles the problem of knowledge graph sparseness.
2. We enable each relation to own different representations for different head and tail entities, which is proved to be helpful to handle the low performance on 1-to-N, N-to-1 and N-to-N relations.
3. We evaluate our TEKE method on different benchmark datasets and experiments demonstrate that TEKE successfully solves the above problems and significantly outperforms state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2 we formally define the problem of text-enhanced knowledge embedding and Section 3 reveals our proposed approach in detail. Section 4 presents the evaluation results and we outline some related work in Section 5. Finally we conclude our work in Section 6.

2 Problem Formulation

In this section, we formally define the problem of text-enhanced knowledge embedding. Here, we first define the input knowledge graph and the text corpus.

A ***knowledge graph*** $\mathcal{KG}$ is a directed graph whose nodes are *entities* and *edges* correspond to the subject-property-object triple facts. Each edge of the form (*head entity*, *relation*, *tail entity*) indicates that there exists a relationship of name *relation* from the head entity to the tail entity, and can be formally represented as (h, r, t), where $h, t \in \mathcal{E}$ are entities and $r \in \mathcal{R}$ is the relation. $\mathcal{E}$ and $\mathcal{R}$ denote the sets of *entities* and *relations* respectively.

A ***text corpus*** (denoted as $\mathcal{D}$) is a set of text documents and can be represented as a sequence of words $\mathcal{D} = \langle w_1 ... w_i ... w_m \rangle$, where w_i denotes the word and m is the length of the word sequence. In our task, it is preferable that the text corpus should cover as many entities in $\mathcal{KG}$ as possible, and the easily accessible Wikipedia pages would be a great choice.

Text-enhanced Knowledge Embedding. Given a $\mathcal{KG}$ and a text corpus $\mathcal{D}$, the text-enhanced knowledge embedding is to learn the entity embeddings $\mathbf{h}, \mathbf{t} \in \mathbb{R}^k$ for each triple (h, r, t) by utilizing the rich text information in $\mathcal{D}$ to deal with low performance on 1-to-N, N-to-1, N-to-N relations and knowledge graph sparseness. In our setting, we also learn the relation embeddings $\mathbf{r} \in \mathbb{R}^k$ following the translation-based methods [Bordes *et al.*, 2013; Wang *et al.*, 2014b; Lin *et al.*, 2015b]. k is the dimension of the learned embeddings (vectors).

3 The Proposed Approach

Given $\mathcal{KG}$ and $\mathcal{D}$, we are to learn the entity and relation embeddings $\mathbf{h}, \mathbf{r}, \mathbf{t}$. As shown in Figure 2, our proposed TEKE contains four key components: (1) **Entity Annotation**: given the text corpus $\mathcal{D}$, we first semantically annotate the entities in $\mathcal{KG}$ by using an entity linking tool automatically. (2) **Textual Context Embedding**: based on the entity-annotated text corpus, we construct a co-occurrence network between entities and words to bridge the KG and text corpus. And then the pointwise and pairwise textual context embeddings are learned. (3) **Entity/Relation Representation Modeling**: we formally formulate the text-enhanced entity/relation embeddings by incorporating the textual context embeddings. (4) **Representation Training**: finally we propose to use a translation-based optimization method to train the entity/relation embeddings. In the following parts, we will describe each component in detail.

3.1 Entity Annotation

Given the text corpus $\mathcal{D} = \langle w_1 ... w_i ... w_m \rangle$, we first use an entity linking tool to automatically label the entities in $\mathcal{KG}$, and get an ***entity-annotated text corpus*** $\mathcal{D}' = \langle x_1 ... x_i ... x_{m'} \rangle$, where x_i corresponds to a word $w \in \mathcal{D}$ or an entity $e \in \mathcal{E}$. We notice that the length m' of $\mathcal{D}'$ is less than the length m of $\mathcal{D}$ because multiple adjacent words could be labeled as one entity.

A general entity linking tool is suitable for this step, e.g. AIDA [Yosef *et al.*, 2011], TAGME [Ferragina and Scaiella, 2010] and Wikify! [Mihalcea and Csomai, 2007]. As we will see in the experiments, we use quite a simple strategy for entity annotation on the Wikipedia text corpus.

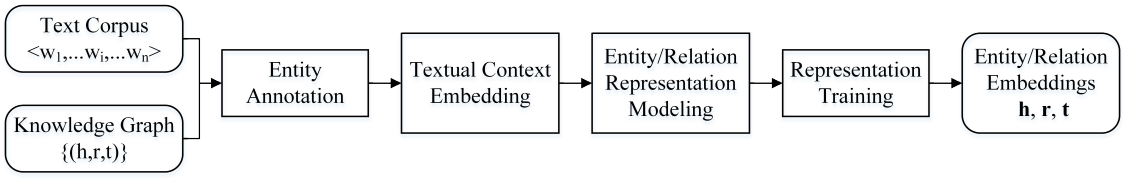

Figure 2: Text-enhanced Knowledge Embedding Framework.

3.2 Textual Context Embedding

In order to bridge the knowledge and text information together, we construct a ***co-occurrence network*** $\mathcal{G} = (\mathcal{X}, \mathcal{Y})$ based on the entity-annotated text corpus $\mathcal{D}'$. $x_i \in \mathcal{X}$ denotes the node of the network and corresponds to a word or an entity. $y_{ij} \in \mathcal{Y}$ represents the co-occurrence frequency between x_i and x_j. The co-occurrence *window* is set to be 5.

The constructed network $\mathcal{G}$ allows us to bridge the entities and words together to utilize the rich text information for representation learning on the knowledge graph. As shown in Figure 1, the neighboring information is helpful to enrich the entity information in the knowledge graph, and we formally define the ***pointwise textual context*** of a given node x_i as its *neighbors*:

$$n(x_i) = \{x_j | y_{ij} > \theta\} \tag{1}$$

where θ is the *threshold* and the neighboring nodes whose co-occurrence frequencies are lower than θ are filtered. For example, $n(\mathbf{Avatar}) = \{\text{film}, \text{movie}, \text{directed}\}$ and $n(\mathbf{James\_Cameron}) = \{\text{director}\}$. Considering that the common neighbors of two nodes could indicate the relationship between them, we define the ***pairwise textual context*** of two nodes as their *common neighbors*:

$$n(x_i, x_j) = \{x_k | x_k \in n(x_i) \cap n(x_j)\} \tag{2}$$

E.g. $n(\mathbf{Avatar}, \mathbf{James\_Cameron}) = \{\text{direct}\}$. On the other hand, we train a word2vec model [Mikolov *et al.*, 2013] on the entity-annotated text corpus $\mathcal{D}'$ by treating each entity as an ordinary word. Thus we get the node representation $\mathbf{x} \in \mathbb{R}^k$ for each node x in $\mathcal{G}$, because the co-occurrence network is directly generated from $\mathcal{D}'$. Based on these representations, we define the ***pointwise textual context embedding*** of x_i as the weighted average of the vectors of the nodes in $n(x_i)$:

$$\mathbf{n}(x_i) = \frac{1}{\sum\limits_{x_j \in n(x_i)} y_{ij}} \sum_{x_j \in n(x_i)} y_{ij} \mathbf{x}_j \tag{3}$$

If $n(x_i)$ is empty, we set $\mathbf{n}(x_i)$ to be $\mathbf{0}$. Similarly, we define the ***pairwise textual context embedding*** of x_i and x_j as the weighted average of the vectors of the nodes in $n(x_i, x_j)$:

$$\mathbf{n}(x_i, x_j) = \frac{1}{Z} \sum_{x_k \in n(x_i, x_j)} \min(y_{ik}, y_{jk}) \mathbf{x}_k \tag{4}$$

where the weight of each common neighbor x_k is set to be the minimum of y_{ik} and y_{jk}, and $Z = \sum_{x_k \in n(x_i, x_j)} \min(y_{ik}, y_{jk})$ is the sum of all weights. If $n(x_i, x_j)$ is empty, we set $\mathbf{n}(x_i, x_j)$ to be $\mathbf{0}$.

3.3 Entity/Relation Representation Modeling

Entity/Relation representation modeling is at the heart of TEKE. Based on the co-occurrence network $\mathcal{G}$ and the learned textual representations, entity/relation representation model is to incorporate the textual context information to the representation learning on knowledge graph. Our representation model is based on the traditional translation-based methods and can be implemented on different optimization targets. Taking TransE [Bordes *et al.*, 2013] as an example, the text-enhanced entity representations $\widehat{\mathbf{h}}$ and $\widehat{\mathbf{t}}$ are defined as the linear transformation of *pointwise textual context embeddings* of h and t.

$$\widehat{\mathbf{h}} = \mathbf{n}(h)\mathbf{A} + \mathbf{h} \tag{5}$$

$$\widehat{\mathbf{t}} = \mathbf{n}(t)\mathbf{A} + \mathbf{t} \tag{6}$$

where $\mathbf{A}$ is a $k \times k$ matrix and can be viewed as the weight of the textual contexts. $\mathbf{h}$, $\mathbf{t}$ could be viewed as the biased vectors. Similarly, the text-enhanced relation representation $\widehat{\mathbf{r}}$ is defined as the linear transformation of the *pairwise textual context embedding* of h and t.

$$\widehat{\mathbf{r}} = \mathbf{n}(h, t)\mathbf{B} + \mathbf{r} \tag{7}$$

where $\mathbf{B}$ is a $k \times k$ weighting matrix and $\mathbf{r}$ could be viewed as the biased vector. The score function is defined as

$$f(h, r, t) = \|\widehat{\mathbf{h}} + \widehat{\mathbf{r}} - \widehat{\mathbf{t}}\|_2^2 \tag{8}$$

By incorporating the textual context embeddings $\mathbf{n}(h)$, $\mathbf{n}(t)$ and $\mathbf{n}(h, t)$, TEKE is better to handle the problem of knowledge graph sparseness. On the other hand, given different pairs of head and tail entities for one particular relation, $\mathbf{n}(h, t)$ owns different representations, which allows the relation to have different representations while holding $\widehat{\mathbf{h}} + \widehat{\mathbf{r}} \cong \widehat{\mathbf{t}}$. Such a property improves the performance on representing 1-to-N, N-to-1 and N-to-N relations.

In practice, we enforce constraints on the norms of the embeddings h, r, t and the weighting matrices, i.e. $\forall h, r, t$, we have $\|\mathbf{h}\|_2 \leq 1$, $\|\mathbf{r}\|_2 \leq 1$, $\|\mathbf{n}(h)\mathbf{A}\|_2 \leq 1$, $\|\mathbf{t}\|_2 \leq 1$, $\|\mathbf{n}(t)\mathbf{A}\|_2 \leq 1$, $\|\mathbf{n}(h, t)\mathbf{B}\|_2 \leq 1$, $\|\widehat{\mathbf{h}}\|_2 \leq 1$, $\|\widehat{\mathbf{r}}\|_2 \leq 1$ and $\|\widehat{\mathbf{t}}\|_2 \leq 1$.

On the other hand, our entity/relation representation model can be easily extended to other knowledge graph representation learning methods. Following TransH [Wang *et al.*, 2014b] and TransR [Lin *et al.*, 2015b], we can further enable an entity to have distinct distributed representations when involved in different relations. Following the idea of TransH, the entity embeddings $\widehat{\mathbf{h}}$ and $\widehat{\mathbf{t}}$ are first projected to the hyperplane of $\mathbf{w}_r$, denoted as $\widehat{\mathbf{h}}_\perp$ and $\widehat{\mathbf{t}}_\perp$. And we can model

the entity representation as $\widehat{\mathbf{h}}_\perp = \widehat{\mathbf{h}} - \mathbf{w}_r^\top \widehat{\mathbf{h}} \mathbf{w}_r$ and $\widehat{\mathbf{t}}_\perp = \widehat{\mathbf{t}} - \mathbf{w}_r^\top \widehat{\mathbf{t}} \mathbf{w}_r$, with the score function $f_\perp(h,r,t) = \|\widehat{\mathbf{h}}_\perp + \widehat{\mathbf{r}} - \widehat{\mathbf{t}}_\perp\|_2^2$. Similarly following the idea of TransR, the entity embeddings $\widehat{\mathbf{h}}$ and $\widehat{\mathbf{t}}$ are first projected to another vector space by the projection matrix $\mathbf{M}_r$, denoted as $\widehat{\mathbf{h}}_r$ and $\widehat{\mathbf{t}}_r$, and we can model the entity representation as $\widehat{\mathbf{h}}_r = \widehat{\mathbf{h}}\mathbf{M}_r$ and $\widehat{\mathbf{t}}_r = \widehat{\mathbf{t}}\mathbf{M}_r$, with the score function $f_r(h,r,t) = \|\widehat{\mathbf{h}}_r + \widehat{\mathbf{r}} - \widehat{\mathbf{t}}_r\|_2^2$.

3.4 Representation Training

We define the training objective as the following margin-based score function

$$L = \sum_{(h,r,t)\in\mathcal{S}} \sum_{(h',r,t')\in\mathcal{S}'} \max(0, f(h,r,t) + \gamma - f(h',r,t'))$$

where $\max(\cdot,\cdot)$ aims to get the maximum of two inputs, γ is the margin, $\mathcal{S}$ is the set of correct triples and $\mathcal{S}'$ is the set of incorrect triples.

Existing knowledge graphs only contain correct triples. It is routine to construct incorrect triples $(h',r,t') \in \mathcal{S}'$ by corrupting correct triples $(h,r,t) \in \mathcal{S}$ with probability-based entity replacement. We follow the strategies used in [Wang *et al.*, 2014b; Lin *et al.*, 2015b] which are denoted as "`unif`" and "`bern`". The learning process is carried out using stochastic gradient descent (SGD). To avoid overfitting, we initialize the entity/relation embeddings with TransE's results, and initialize all matrices as identity matrices.

4 Experiments and Analysis

4.1 Experimental Setup and Datasets

For the knowledge graphs to be represented, we employ several datasets commonly used in previous methods, which are generated from WordNet [Miller, 1995] and Freebase [Bollacker *et al.*, 2008]. WordNet is a large lexical database of English with each entity as a synset which is consisting of several words and corresponds to a distinct word sense. Freebase is a large knowledge graph of general world facts. Following [Bordes *et al.*, 2013; Wang *et al.*, 2014b; Lin *et al.*, 2015b; Socher *et al.*, 2013], we adopt four benchmark datasets for evaluation, which are WN18 and WN11 generated from WordNet, FB15K and FB13 generated from Freebase. The detailed statistics of the datasets are shown in Table 1.

Table 1: Statistics of the data sets.

Dataset	#$\mathcal{R}$	#$\mathcal{E}$	#Triples(Train/Valid/Test)		
WN18	18	40,943	141,442	5,000	5,000
FB15K	1,345	14,951	483,142	50,000	59,071
WN11	11	38,696	112,581	2,609	10,544
FB13	13	75,043	316,232	5,908	23,733

The text corpus is generated from the English Wikipedia dump archived in August 2015. We remove the documents of those entities whose titles contain one of the following strings: *wikipedia*, *wikiprojects*, *lists*, *mediawiki*, *template*, *user*, *portal*, *categories*, *articles*, *pages*, and *by*, and get 4,919,463 documents in total. As mentioned in Section 3, to incorporate the text information we need to first annotate the entities in the knowledge graphs. For FB15K and FB13, we just focus on the Wikipedia inner links and automatically annotate the links as the Freebase entities if the linked Wikipedia entities have the same titles as the Freebase entities, otherwise as the lexical words. For WN18 and WN11, we ignore the Wikipedia links and annotate the words as the WordNet entities if the words belong to the WordNet synsets. Further, we remove the stop words and the words occurring less than 5 times, and apply word stemming on the entity-annotated texts. We train the skip-gram word2vec model [Mikolov *et al.*, 2013] on the entity-annotated texts. Table 2 shows the detailed statistics of the entity-annotated text corpuses for each benchmark knowledge graph, including the number of annotated entities and the number of distinct word stems.

Table 2: Statistics of entity-annotated Wikipedia corpuses.

KG	#Entities	#Annotated Entities	#Word Stems
WN18	40,943	32,249	1,529,251
FB15K	14,951	14,405	744,983
WN11	38,696	30,937	1,526,467
FB13	75,043	69,208	706,484

We implement different TEKE methods based on different translation-based model, which are TransE, TransH and TransR (see Section 3.3). For simplicity, we denote our text-enhanced methods as **TEKE_E**, **TEKE_H** and **TEKE_R** accordingly. Following [Bordes *et al.*, 2013; Wang *et al.*, 2014b; Lin *et al.*, 2015b], we empirically evaluate TEKE methods on two tasks: link prediction and triple classification. Especially, we will testify TEKE's capability to handle low performance on 1-to-N, N-to-1 and N-to-N relations and knowledge graph sparseness in the task of link prediction.

4.2 Link Prediction

Link prediction is to predict the missing entity h or t for a relation fact triple (h,r,t). For each missing entity, this task is to give a ranking list of candidate entities from the knowledge graph, rather than just giving the best answer. Following [Bordes *et al.*, 2013; Wang *et al.*, 2014b; Lin *et al.*, 2015b], we conduct our experiments using the datasets WN18 and FB15K.

Evaluation protocol. For each testing triple (h,r,t), we replace the head/tail entity by every entity in the knowledge graph, and rank these entities in descending order of the scores calculated by score function f (or $f_\perp$, f_r). Based on the entity ranking lists, we use two evaluation metrics by aggregating over all testing triple: (1) the averaged rank of correct entities (denoted as *Mean Rank*); (2) the proportion of ranks no larger than 10 (denoted as *Hits@10*). Notice that a corrupted triple may also exist in the knowledge graphs, such a prediction should be considered as correct. However, the above evaluations do not deal with the issue and may underestimate the results. To eliminate this factor, we filter out those corrupted triples which appear in either training, validation or testing sets before getting the ranking lists. We name the first evaluation setting as "`Raw`" and the second one as

Table 3: Experimental Results on Link Prediction.

Datasets	WN18				FB15K			
Metric	Mean Rank		Hits@10 (%)		Mean Rank		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE / TEKE_E	263 / **140**	251 / **127**	75.4 / **80.0**	89.2 / **93.8**	243 / **233**	125 / **79**	34.9 / **43.5**	47.1 / **67.6**
TransH / TEKE_H unif	318 / **142**	303 / **128**	75.4 / **79.7**	86.7 / **93.6**	**211** / 228	84 / **75**	42.5 / **44.9**	58.5 / **70.4**
TransH / TEKE_H bern	401 / **127**	388 / **114**	73.0 / **80.3**	82.3 / **92.9**	212 / **212**	**87** / 108	45.7 / **51.2**	64.4 /**73.0**
TransR / TEKE_R unif	232 / **203**	219 / **203**	78.3 / **78.4**	91.7 / **92.3**	**226** / 237	**78** / 79	43.8 / **44.3**	65.5 / **68.5**
TransR / TEKE_R bern	238 / **197**	225 / **193**	**79.8** / 79.4	**92.0** / 91.8	**198** / 218	**77** / 109	48.2 / **49.7**	68.7 / **71.9**

Table 4: Experimental Results on FB15K by Mapping Properties of Relations. (%)

Tasks	Prediction Head (Hits@10)				Prediction Tail (Hits@10)			
Relation Category	1-to-1	1-to-N	**N-to-1**	**N-to-N**	1-to-1	**1-to-N**	N-to-1	**N-to-N**
TransE/TEKE_E	43.7 / **48.9**	65.7 / **72.1**	18.2 / **52.3**	47.2 / **76.8**	43.7 / **46.3**	19.7 / **50.2**	66.7 / **75.3**	50.0 / **76.1**
TransH/TEKE_H unif	**66.7** / 66.6	**81.7** / 80.9	30.2 / **58.0**	57.4 / **79.6**	**63.7** / 60.5	30.1 / **60.4**	**83.2** / 81.5	60.8 / **80.2**
TransH/TEKE_H bern	66.8 / **69.3**	87.6 / **90.8**	28.7 / **54.1**	64.5 / **82.0**	**65.5** / 60.7	39.8 / **61.5**	83.3 / **88.3**	67.2 / **82.1**
TransR/TEKE_R unif	**76.9** / 66.2	77.9 / **82.0**	38.1 / **57.0**	66.9 / **81.3**	**76.2** / 62.5	38.4 / **57.5**	76.2 / **83.1**	69.1 / **81.2**
TransR/TEKE_R bern	**78.8** / 70.1	89.2 / **89.3**	34.1 / **54.0**	69.2 / **81.7**	**79.2** / 69.6	37.4 / **59.2**	**90.4** / 89.2	72.1 / **83.5**

"Filter". In both settings, a lower *Mean Rank* is better while a higher *Hits@10* is better.

Implementation. As the datasets are the same, we directly compare our methods with the baselines reported in [Lin *et al.*, 2015b], where the translation-based methods including TransE, TransH and TransR achieved the state-of-the-art performance. We set the neighboring threshold θ on the co-occurrence network to be 10, and select learning rate λ for SGD among $\{0.1, 0.01, 0.001\}$, the margin γ among $\{1, 2, 4\}$, the embedding dimension k among $\{20, 50, 100\}$, the batch size B among $\{120, 1440, 4800\}$. The best configuration is determined according to the mean rank in validation set. We traverse all the training triples for 1,000 times.

Result analysis. The results are reported in Table 3. We observe that: (1) On WN18, TEKE methods perform much better than the baselines in terms of *Mean Rank*. One reason may be that WordNet itself is a lexical database and the difference between WordNet and the text corpus is quite small. On the other hand, no much improvement is observed on FB15K in terms of *Mean Rank*. One reason is that the *Mean Rank* is easily reduced by an obstinate triple with a low rank. Another reason is that our proposed methods aim to better handle 1-to-N, N-to-1 and N-to-N relations, and thus tend to give multiple entities a higher rank simultaneously, which results in a lower rank for the target entity in the testing triple. (2) On both WN18 and FB15K, TEKE methods outperform other baselines significantly and consistently in terms of *Hits@10*, which indicates the effectiveness of incorporating text information to knowledge graph representation learning. (3) TransR performs slightly better than TransH, while TEKE_H performs slightly better than TEKE_R. The reason could be that TEKE_R incorporates more parameters and it needs more training rounds to convergence than TEKE_H.

Capability to handle 1-to-N, N-to-1 and N-to-N relations. Table 4 shows separate evaluation results by mapping properties of relations on FB15K. Following [Bordes *et al.*, 2013], we divide relations into four types: 1-to-1, 1-to-N, N-to-1 and N-to-N, for which the proportions in FB15K are 24.2%, 22.9%, 28.9% and 24.0% respectively, based on the measure used in [Wang *et al.*, 2014b]. TEKE methods significantly outperform the baselines when predicting the entity where multiple entities could be correct, which means to predict head entities in N-to-1 and N-to-N relations, and to predict tail entities in 1-to-N and N-to-N relations. The averaged improvement achieves about 20%, which indicates the capability of our methods to handle low performance on the 1-to-N, N-to-1 and N-to-N relations. On the other side, TEKE methods have not shown much advantage for predicting the entity where only one entity is correct, which means to predict heads in 1-to-1 and 1-to-N relations, and to predict tails in 1-to-1 and N-to-1 relations. The reason is that our proposed methods aim to better handle 1-to-N, N-to-1 and N-to-N relations, and thus tend to give multiple entities a higher rank simultaneously, which results in a lower rank for the target entity. Nevertheless, as shown in Table 3, the overall performance on all relations is still better than the baselines.

Capability to handle knowledge graph sparseness. As mentioned in Section 1, we conduct a case study to reveal the influence of KG structure sparseness and TEKE's capability to handle the problem. We randomly select 3,000 entities and the associated triples from FB15K, and get the FB3K dataset with 2,238 testing triples and 2,106 validation triples. Based on FB3K, we further randomly select 3,000 entities and the associated triples from FB15K and get the FB6K dataset, and get FB9K dataset based on FB6K by using the same strategy. Table 5 shows the detailed statistics of the datasets, where the number of triples per entity ($\#\mathcal{T}/\#\mathcal{E}$) and the number of triples per relation ($\#\mathcal{T}/\#\mathcal{R}$) reveal the graph densities.

Table 5: Datasets with different densities.

Dataset	$\#\mathcal{E}$	$\#\mathcal{R}$	$\#\mathcal{T}$	$\#\mathcal{T}/\#\mathcal{E}$	$\#\mathcal{T}/\#\mathcal{R}$
FB3K	3,000	613	19,339	6.45	31.55
FB6K	6,000	913	75,347	12.56	82.53
FB9K	9,000	1,094	167,191	18.58	152.83

$\mathcal{T}$ represents the training triples.

For a fair comparison, we evaluate *Mean Rank* on the same testing dataset, i.e. ranking 3,000 entities for 2,238 triples for all three datasets. As shown in Table 6, we observe that TEKE_E outperforms TransE on different graph sparseness levels. As the graph density gets higher, both TransE and TEKE_E perform better (with a lower rank). Besides, TEKE_E achieves the highest improvement on the sparsest FB3K dataset, and achieves comparable results on FB6K to TransE on FB9K, which shows TEKE's capability to deal with the graph structure sparseness.

Table 6: Mean Rank Comparison.

Methods	TransE / TEKE_E			
Metric	Raw		Filter	
FB3K	102.7	**94.9**	41.7	**34.8**
FB6K	81.9	**78.1**	29.8	**25.6**
FB9K	79.5	**77.0**	27.6	**24.7**

4.3 Triple Classification

Triple classification is to judge whether a given triple (h, r, t) is correct or not. It is a binary classification task which has been widely explored in [Socher *et al.*, 2013; Bordes *et al.*, 2013; Wang *et al.*, 2014b; Lin *et al.*, 2015b]. Following [Socher *et al.*, 2013], we conduct our experiments using WN11 and FB13, which already contain negative triples obtained by corrupting correct triples.

Evaluation protocol. For each triple (h, r, t), if the score obtained by score function f (or $f_\perp$, f_r) is below a relation-specific threshold δ_r, the triple will be classified as positive, otherwise as negative. The threshold δ_r is optimized by maximizing classification accuracies on the validation set.

Implementation. As the datasets are the same, we directly compare our methods with baselines reported in [Lin *et al.*, 2015b]. The parameter settings are the same as those in the task of link prediction. The best configuration is determined according to the accuracy in validation set. We traverse all the training triples for 1000 times.

Result analysis. The results are reported in Table 7. We observe that: (1) On both WN11 and FB13, TEKE_E and TEKE_H consistently outperform the comparison methods, especially on WN11. The reason may be that WordNet itself is a lexical database and the difference between WordNet and the text corpus is quite small. (2) TEKE_R's unif implementation on WN11 and bern implementation on FB13 perform better than TransR, while the other implementations perform a bit worse. The reason could be that TEKE_R incorporates more parameters and it needs more training rounds than 1,000 to convergence.

5 Related Work

Existing translation-based learning methods learn the entity embeddings directly from the graph structure between entities. TransE [Bordes *et al.*, 2013] treats the relations as translation operations from head entity to tail entity, and wants $\mathbf{r} \approx \mathbf{t} - \mathbf{h}$ when (h, r, t) holds. TransE applies well to 1-to-1 relations but has issues for 1-to-N, N-to-1 and N-to-N relations. For example $\forall i \in \{0, ..., m\}$, $(h, r, t_i) \in \mathcal{S}$, we will get the same representations for those different entities $t_0, ..., t_m$. TransH [Wang *et al.*, 2014b] and TransR [Lin *et al.*, 2015b] enable an entity to have different representations for different relations by preliminarily generating the relation-specific entity embeddings with mathematical transformations, which are hyperplane projection for TransH and space projection for TransR. PTransE is a multiple-step relation path-based representation learning model proposed in [Lin *et al.*, 2015a]. As shown in Section 4.2, the performance on 1-to-N, N-to-1 and N-to-N relations is still unsatisfactory. On the other side, by directly learning the embeddings from the graph structure, the performance is limited due to KG sparseness which is quite common especially in the domain-specific and non-English situations.

Table 7: Evaluation results of triple classification. (%)

Datasets	WN11	FB13
TransE / TEKE_E `unif`	75.9 / **84.1**	70.9 / **75.1**
TransE / TEKE_E `bern`	75.9 / **84.5**	81.5 / **82.1**
TransH / TEKE_H `unif`	77.7 / **84.3**	76.5 / **77.4**
TransH / TEKE_H `bern`	78.8 / **84.8**	83.3 / **84.2**
TransR / TEKE_R `unif`	**85.5** / 85.2	74.7 / **77.1**
TransR / TEKE_R `bern`	85.9 / **86.1**	**82.5** / 81.6

There are also several methods incorporating textual information to improve representation learning of KG. [Socher *et al.*, 2013] proposes a neural tensor network method by representing an entity as the average of its word embeddings in entity name, which allows the sharing of textual information located in similar entity names. [Wang *et al.*, 2014a] combines entity embeddings with word embeddings into a joint continuous vector space by alignment models using entity names or Wikipedia anchors. Considering the entity names are usually short and ambiguous, several methods are proposed to further improve the performance by utilizing the entity descriptions. [Zhong *et al.*, 2015] extends the joint model of [Wang *et al.*, 2014a] and aligns knowledge and text embeddings by entity descriptions. [Zhang *et al.*, 2015] represents entities with entity names or the average of word embeddings in descriptions. [Xie *et al.*, 2016] proposes a description-embodied knowledge representation learning method which learns the entity embedding by both modeling the corresponding fact triples and the description. However, lots of entity descriptions are not available in practical KGs.

Inspired by the idea of distant supervision, our TEKE methods take a text corpus as input and attempt to incorporating deep contextual information to the KG. To the best of our knowledge, TEKE is the first text-associated method to deal with the problem of low performance on 1-to-N, N-to-1 and N-to-N relations by enabling each relation to own different representations for different head/tail entities.

6 Conclusion and Future Work

In this paper, we propose a novel representation learning method named TEKE for KG. Our TEKE methods better handle the problems of low performance on 1-to-N, N-to-1 and N-to-N relations and KG sparseness. In our future work, we will concentrate on further improving the performance on 1-to-1 relations and try to incorporate the knowledge reasoning

process into the representation learning.

Acknowledgments

The work is supported by 973 Program (No. 2014CB340504), NSFC-ANR (No. 61261130588), NSFC key project(No. 61533018), Tsinghua University Initiative Scientific Research Program (No. 20131089256), and THU-NUS NExT Co-Lab. Besides, we gratefully acknowledge the assistance of Zhiyuan Liu (Tsinghua University) and Jie Tang (Tsinghua University) for improving the paper work.

References

[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 *SIGMOD*, pages 1247–1250, 2008.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

[Ferragina and Scaiella, 2010] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*, pages 1625–1628, 2010.

[Lin *et al.*, 2015a] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *EMNLP*, pages 705–714, 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 *AAAI*, pages 2181–2187, 2015.

[Mihalcea and Csomai, 2007] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, 2007.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Chen Kai, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Miller, 1995] G. A. Miller. WordNet: a Lexical Database for English. *Communications of the ACM*, 38:39–41, 1995.

[Mintz *et al.*, 2009] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL/IJCNLP*, pages 1003–1011, 2009.

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

[Wang *et al.*, 2013] Zhigang Wang, Zhixing Li, Juanzi Li, Jie Tang, and Jeff Z. Pan. Transfer learning based cross-lingual knowledge extraction for wikipedia. In *ACL*, pages 641–650, 2013.

[Wang *et al.*, 2014a] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In *EMNLP*, pages 1591–1601, 2014.

[Wang *et al.*, 2014b] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119, 2014.

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

[Yosef *et al.*, 2011] Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. Aida: An online tool for accurate disambiguation of named entities in text and tables. In *PVLDB*, volume 4, pages 1450–1453, 2011.

[Zhang *et al.*, 2015] Dongxu Zhang, Bin Yuan, Dong Wang, and Rong Liu. Joint semantic relevance learning with text data and graph knowledge. In *ACL-IJCNLP*, pages 32–40, 2015.

[Zhong *et al.*, 2015] Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. Aligning knowledge and text embeddings by entity descriptions. In *EMNLP*, pages 267–272, 2015.