Text-aware Knowledge Graph Embedding with Jointly Neighbor Contexts and Textual Descriptions

# 1212

ML: Knowledge-based Learning
NLP: NLP Applications and Tools

Abstract

This paper focuses on proposing an approach to improving text-aware knowledge graph embedding by combining neighbor contexts and textual descriptions. Like textual description, neighbor context also characterizes and supplements the semantic properties of an entity from a different aspect. However, each entity has a different number of neighbors and the semantic connections between them are diverse. Therefore, we should consider how to embed the neighbor contexts of entities and then combine the embeddings of neighbor contexts with textual descriptions together. By introducing the topic model to embed the neighbor context, we propose a novel text-aware model named Joint Neighbor Contexts and Textual Descriptions (JNCTD), the composition methods of which are weighted composition and projection composition. We evaluate our model with two tasks including link prediction and entity classification on two benchmark datasets WN18 and FB15K. Experimental results show that our model consistently outperforms other baselines.

1 Introduction

Knowledge graph has been proved to benefit many artificial intelligence applications, such as question answering, *etc.* Knowledge graph consists of many *triple facts* $(head\ entity, relation, tail\ entity)$, denoted as (h, r, t), indicating the *relation* between two *entities*. However, it is impossible to collect all relations among entities in knowledge graphs and as result knowledge graphs often suffer from incompleteness. Therefore, *knowledge graphs completion*, which aims at completing missing triples, *i.e.*, predict t given (h, r) or predict h given (r, t), is of great significance. Knowledge graph embedding is a key technique in knowledge graphs completion and embedding methods can be categorized into three branches: *triple-only embedding*, *graph-context-aware embedding*, and *text-aware embedding*.

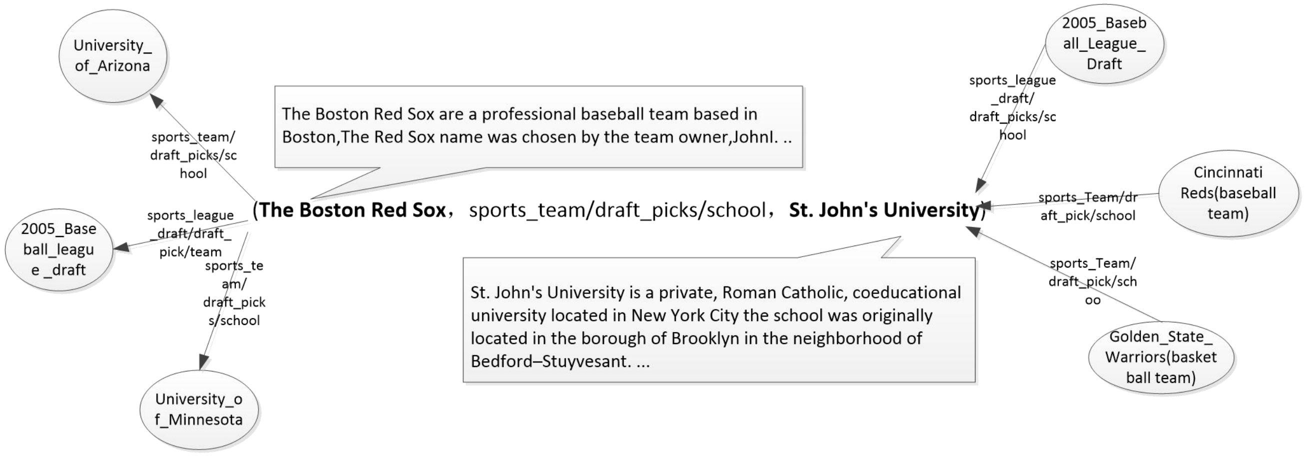
Triple-only embedding models only consider symbolic triples. TransE [Bordes *et al.*, 2013] is a typical model which translates the *head* entity to the *tail* one by the relation vector, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, indicating that the tail embedding $\mathbf{t}$ should be the nearest neighbor of $\mathbf{h} + \mathbf{r}$. Other methods model entities and relations in different ways, including TransH [Wang *et al.*, 2014], TransR [Lin *et al.*, 2015b], TransG [Xiao *et al.*, 2016], TransD [Ji *et al.*, 2015], TranSparse [Ji *et al.*, 2016], and KG2E [He *et al.*, 2015], *etc.* Graph-context-aware embedding utilizes graph context structure information to better embed the entities and relations, such as PTransE [Lin *et al.*, 2015a] and TCE [Shi *et al.*, 2017]. Text-aware embedding has been proposed to learn knowledge graph embeddings by utilizing entities related textual descriptions information. [Zhong *et al.*, 2015] applied a joint model to align entities and textual embeddings. DKRL [Xie *et al.*, 2016] adopts CNN-structure to represent textual descriptions and combines with entities embeddings. SSP [Xiao *et al.*, 2017] utilizes the *topic model* to process textual descriptions information and unites entities embeddings with a projection method.

However, as far as the text-aware embedding is concerned, we observe that the semantic correlations of an entity depend on the *neighbor* entities and its textual descriptions. If there is a large number of neighbor entities, we should consider much more on its neighbor entities rather than textual descriptions. Therefore, this motivates us to consider text-aware knowledge embedding with neighbor contexts and textual descriptions. We illustrate our motivation via the following example selected from Freebase.

As showed in Figure 1, the triple $(The\ Boston\ Red\ Sox$, $sports\_team/draft\_picks/school, St.John's\ University)$ whose head entity $The\ Boston\ RedSox$ is a baseball team and tail entity $St.John's\ University$ is a university. And the textual description of either entity does not characterize the semantic connections between them clearly. However, the head entity $The\ Boston\ RedSox$ has neighbor entities like $University\_of\_Arizona$ (university) and $University\_of\_MLnnesota$ (university) which are closely related to the tail entity $St.John's\ University$. Analogously, the tail entity neighbors like $Cincinnati\_Reds$ (baseball team) and $Golden\_State\_Warriors$ (basketball team) are also closely related to the head entity. Therefore, we should design a new text-aware model jointly learned from both neighbor contexts and textual descriptions.

However, there are two problems with this approach:

1. The number of each entity neighbors in a knowledge graph is different. Furthermore, different neighbor contexts represent diverse semantics. So we should design



The Boston Red Sox are a professional baseball team based in Boston,The Red Sox name was chosen by the team owner,JohnI. ..

St. John's University is a private, Roman Catholic, coeducational university located in New York City the school was originally located in the borough of Brooklyn in the neighborhood of Bedford–Stuyvesant. ...

Figure 1: An triple example with entities textual descriptions and neighbor contexts in Freebase FB15K.

an appropriate method to process and embed the neighbor contexts of entities.

2. Even though the neighbor context and the textual description of an entity show different characteristics, we should design a joint model to combine the embeddings of neighbor contexts and textual descriptions.

To tackle the above problems, we propose a novel text-aware model named *Joint Neighbor Context and Textual Descriptions (JNCTD)*. For the first problem, the entities are usually organized by topics in knowledge graphs, for example, *entity type* has been used in Freebase to categorize entities [Xiao *et al.*, 2017]. Therefore, we adopt a topic model [Blei, 2012], which can mine topic level information correlated well with human concepts without any supervision, to process and embed the neighbor context. For the second problem, in order to combine neighbor contexts and textual descriptions, we propose two composition methods which are *weighted composition* and *projection composition*. We evaluate our model with two tasks including link prediction and entity classification on two benchmark datasets that are the subsets of Wordnet [Miller, 1995] and Freebase [Bollacker *et al.*, 2008]. Experimental results show that our model with either weighted composition or projection composition consistently outperforms other baselines, including DKRL [Xie *et al.*, 2016] and SSP [Xiao *et al.*, 2017].

2 Preliminary

In this section, we recall the notations and core functions of related models including topic model NMF and three translation-based models TransE, TransH, and SSP.

2.1 Topic Model

Stevens *et al.* [2012] presented a simple and effective topic model, Non-negative Matrix Factorization (NMF). Topic model extracts and represents relations between words and documents. The core function of NMF can be stated as:

$$\mathcal{L}_{topic} = \sum_{doc \in DOC, w \in \mathcal{W}_d} (\mathcal{C}_{doc,w} - \mathbf{s}_{doc}^{\top}\mathbf{w})^2, \quad (1)$$

where DOC is a set of documents, doc is a document in DOC, and $\mathcal{W}_d$ is a set of words of the doc. $\mathcal{C}_{doc,w}$ is the times of the word w occurring in the document doc. $\mathbf{s}_{doc}$ is a semantic vector of document doc and $\mathbf{w}$ is a topic distribution of word w. The stochastic gradient descent algorithm (SGD) is adopted to minimize the loss $\mathcal{L}_{topic}$. By training this objective function, we could get a semantic vector of s_{doc} for each document.

2.2 Translation-based Models

Translation-based models embed entities into a vector space and enforce the embeddings compatible under a score function. Here we introduce TransE [Bordes *et al.*, 2013] and TransH [Wang *et al.*, 2014] in detail as we extend some ideas of them in our model. We denote vectors by bold lower case letters like $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$. Score function is represented by $f(h, r, t)$.

TransE. For each triple (h, r, t), TransE [Bordes *et al.*, 2013] wants $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds. This indicates that $\mathbf{t}$ should be the nearest entity from $(\mathbf{h} + \mathbf{r})$. Hence, TransE defines the following score function:

$$f_{trE}(h, r, t) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||_{L1/L2} \quad (2)$$

And the function returns low score if (h, r, t) holds, vice versa.

TransH. TransH [Wang *et al.*, 2014] enables an entity to have distinct embeddings when involved in different relations. For a relation r, TransH models the relation with a vector $\mathbf{r}$ and a hyperplane with $\mathbf{w}_r$ as a normal vector. Then the score function of TransH is defined as

$$\mathbf{h}_{\perp} = \mathbf{h} - \mathbf{w}_{\mathbf{r}}^{\top}\mathbf{h}\mathbf{w}_{\mathbf{r}} \quad (3)$$

$$\mathbf{t}_{\perp} = \mathbf{t} - \mathbf{w}_{\mathbf{r}}^{\top}\mathbf{t}\mathbf{w}_{r} \quad (4)$$

$$f_{trH}(h, r, t) = ||\mathbf{h}_{\perp} + \mathbf{r} - \mathbf{t}_{\perp}||_{L1/L2} \quad (5)$$

SSP. SSP [Xiao *et al.*, 2017] utilizes topic model to process textual descriptions information, generate semantic spaces, and project symbolic triples losses to semantic spaces, which jointly learns embeddings from symbolic triples and textual descriptions. SSP considers textual descriptions of entities as documents. So we can get a semantic vector for each entity through Equation 1. $\mathbf{d}_h$ and $\mathbf{d}_t$ are semantic vectors from head and tail entities textual descriptions respectively. They are combined in addition form and represented as follows:

$$\mathbf{s} = \frac{\mathbf{d}_h + \mathbf{d}_t}{||\mathbf{d}_h + \mathbf{d}_t||_2^2} \tag{6}$$

The loss function of SSP is defined as follows:

$$f_{SSP}(h, r, t) = \lambda||\mathbf{L} - \mathbf{s}^\top \mathbf{L}\mathbf{s}||_2^2 + ||\mathbf{L}||_2^2, \tag{7}$$

where $\mathbf{L} = f_{trE}(h, r, t)$ is the loss from Equation 2 and $\mathbf{L} - \mathbf{s}^\top \mathbf{L}\mathbf{s}$ is the component of loss $\mathbf{L}$ on the semantic hyperplane. λ is introduced to balance the two parts.

3 Methodology

In this section, to embed neighbor contexts of entities and combine the embeddings of neighbor context and textual descriptions of entities, we propose a text-aware model named *Joint Neighbor Context and Textual Descriptions (JNCTD)*.

3.1 Neighbor Context

We first introduce some notations. Let $\mathcal{K}$ be a knowledge graph. Each *triple* in $\mathcal{K}$ is denoted as (h, r, t), in which h denotes head entity, t denotes tail entity, and r denotes the relation between h and t. E is a set of all entities in $\mathcal{K}$ and R is a set of all relations in $\mathcal{K}$. $\mathbf{d}_h$ and $\mathbf{d}_t$ are the head-specific and tail-specific semantic vectors generated from the entities textual descriptions using Equation 1. Similarly, $\mathbf{n}_h$ and $\mathbf{n}_t$ are the semantic vectors generated from head neighbor context and tail neighbor context using Equation 1 respectively.

The neighbor context of an entity is the surroundings of it in the knowledge graph. It is the structural information that interacts most with entities. Given an entity e, the neighbor context of e consists of its directed linked entities including head and tail neighbors. Specifically, the head neighbors is a set $\varepsilon_{head}(e) = \{h|(h, r, e) \in \mathcal{K}\}$ and the tail neighbors is a set $\varepsilon_{tail}(e) = \{t|(e, r, t) \in \mathcal{K}\}$. Therefore, the neighbor context of e is a set $\varepsilon_N(e) = \varepsilon_{head}(e) \cup \varepsilon_{tail}(e)$.

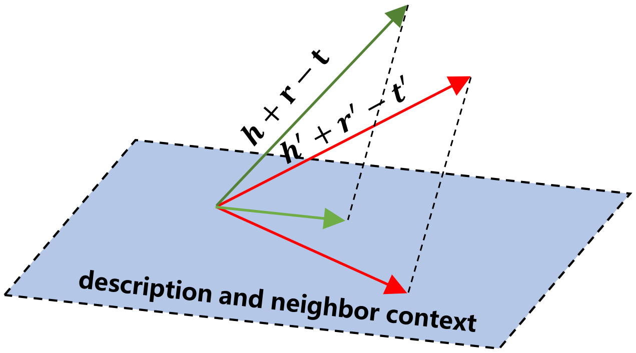
3.2 Semantic Vector Composition

To combine neighbor contexts and textual descriptions jointly, we propose and test two joint methods which are *weighted composition* and *projection composition*.

Weighted Composition

As either neighbor contexts or textual descriptions can be considered as the attributes of entities, the topics from each are generally the same. Furthermore, for a specific entity, these two parts of neighbor context and textual description can be complementary to each other. So we adopt an addition form to collect as complete as possible semantic topic correlations between entities. Then we generate a semantic normal vector by addition form and use it to project the loss into a semantic hyperplane. The overview of the weighted composition is shown in Figure 2.

In order to align the topics of each dimension of the semantic vectors from neighbor contexts and textual descriptions, we conduct the two topic models and the embedding model simultaneously. Moreover, we introduce a weight parameter β in $[0, 1]$ to balance the two parts. So the semantic vector of head entity is formally represented as:



$\mathbf{h}+\mathbf{r}-\mathbf{t}$
$\mathbf{h}'+\mathbf{r}'-\mathbf{t}'$
description and neighbor context

Figure 2: Overview of weighted composition. $\mathbf{h}+\mathbf{r}-\mathbf{t}$ is the loss vector of positive triple and $\mathbf{h}'+\mathbf{r}'-\mathbf{t}'$ is the negative loss vector. They are equal in length so it is hard to identify correctness. But project them into the semantic hyperplane by a normal vector generated from neighbor context and textual description, they are easy to distinguish.

$$\mathbf{s}_h = \beta\mathbf{d}_h + (1-\beta)\mathbf{n}_h \tag{8}$$

Similarly, the representation of tail entity semantic vector is:

$$\mathbf{s}_t = \beta\mathbf{d}_t + (1-\beta)\mathbf{n}_t \tag{9}$$

If entities with few neighbors, the correlations between entities may largely be determined by textual descriptions. But if entities with many neighbors, the correlations depend more on their neighbor contexts. Therefore, we define β as:

$$\beta = \frac{1}{\frac{n^k}{a} + 1}, \tag{10}$$

where n is the number of each entity neighbors, k and a are hyper-parameters. When n is relatively large we will consider more about textual description otherwise, we will take much more into account the neighbor context.

As each dimension of a semantic vector indicates the relevant level to a topic, we also adopt addition form like Equation 6 to combine the head and tail semantic vectors:

$$\mathbf{s} = \frac{\mathbf{s}_h + \mathbf{s}_t}{||\mathbf{s}_h + \mathbf{s}_t||_2^2} \tag{11}$$

Then we follow the Equation 3 to project the loss vector $\mathbf{L}$ into the semantic hyperplane by the normal vector $\mathbf{s}$:

$$\mathbf{L}_s = \mathbf{L} - \mathbf{s}^\top \mathbf{L}\mathbf{s} \tag{12}$$

Projection Composition

Neighbor contexts and textual descriptions represent the attributes of entities from two different aspects. Therefore, we use a quadratic projection method to get a more accurate projection component. We project the loss into the neighbor context and textual description one after another. The overview of this method is shown in Figure 3.

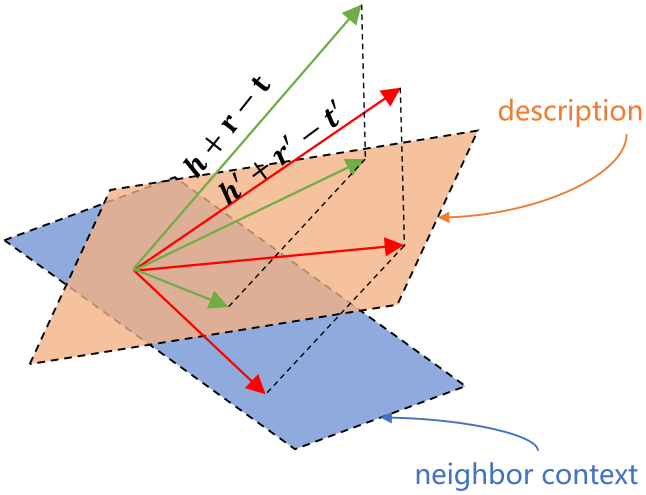

$\mathbf{h}+\mathbf{r}-\mathbf{t}$
$\mathbf{h}'+\mathbf{r}'-\mathbf{t}'$
description
neighbor context

Figure 3: Overview of projection composition. When projecting to the textual description hyperplane, they are still difficult to distinguish. And then we continue to project them into the neighbor context hyperplane.

According to our previous analysis, we first use the Equation 6 to combine head and tail semantic vectors including from textual descriptions and neighbor contexts:

$$\mathbf{s}_d = \frac{\mathbf{d}_h + \mathbf{d}_t}{||\mathbf{d}_h + \mathbf{d}_t||_2^2}, \quad \mathbf{s}_n = \frac{\mathbf{n}_h + \mathbf{n}_t}{||\mathbf{n}_h + \mathbf{n}_t||_2^2}, \tag{13}$$

where $\mathbf{s}_d$ and $\mathbf{s}_n$ are the unit normal vectors of textual description and neighbor context hyperplanes respectively.

Then we follow Equation 3 to project the loss $\mathbf{L}$ to $\mathbf{L}_{ds}$ which is in the textual description hyperplane:

$$\mathbf{L}_{ds} = \mathbf{L} - \mathbf{s}_d^\top \mathbf{L}\mathbf{s}_d \tag{14}$$

Since the textual descriptions of entities might be not always precise or closely related to knowledge graph, sometimes it is still difficult to distinguish the positive triples and the negative triples in the textual description hyperplane. As a result, we continue to project $\mathbf{L}_{ds}$ to the neighbor context hyperplane:

$$\mathbf{L}_s = \mathbf{L}_{ds} - \mathbf{s}_n^\top \mathbf{L}_{ds}\mathbf{s}_n \tag{15}$$

3.3 Objective and Training

We consider respectively embedding-specific $\mathbf{L}$ and semantic-specific $\mathbf{L}_s$ in training objective. In order to balance these two parts, a hyper-parameter λ is introduced. Then the total loss function of our model is as following:

$$f_{JNCTD}(h, r, t) = \lambda||\mathbf{L}_s||_2^2 + ||\mathbf{L}||_2^2 \tag{16}$$

To encourage discrimination between positive and negative triples, we use the following margin-based ranking loss:

$$\sum_{(h,r,t)\in\mathcal{K}} \sum_{(h',r',t')\in\mathcal{K}'} [f_{JNTCD}(h,r,t) + \gamma - f_{JNTCD'}(h',r',t')]_+, \tag{17}$$

where $\gamma > 0$ is a margin separating positive and negative triples, $[x]_+ = max(x, 0)$ is the hinge loss, $\mathcal{K}'$ is a negative triples set sampled with the *Bernoulli sampling method* introduced in [Wang *et al.*, 2014] as following:

$$\begin{aligned}\mathcal{K}' = \{(h', r, t)|h' \in E\} \cup \{(h, r, t')|h' \in E\} \\ \cup\{(h, r', t)|h' \in R\},\end{aligned} \tag{18}$$

where E and R are the sets of entities and relations in $\mathcal{K}$ respectively. In $\mathcal{K}'$ the head, tail, or relation of a triple are randomly replaced by another entity in E or relation in R.

For aligning the semantic vectors, we conduct the two topic-specific models and the embedding-specific model simultaneously. Overall, the total loss is:

$$\mathcal{L} = \mathcal{L}_{JNCTD} + \mu\mathcal{L}_{topic}, \tag{19}$$

where μ is introduced to balance the two parts.

We initialize the embedding vectors by the similar methods used in the deep neural network [Glorot and Bengio, 2010] and adopt the stochastic gradient descent algorithm (SGD) in the optimization to minimize the loss.

4 Experiment and Analysis

In this section, we evaluate our model with two typical text-aware embedding tasks: link prediction and entity classification. We first introduce our datasets and then analyze the experimental results in detail.

4.1 Datasets

In this paper, we adopted subsets of Wordnet [Miller, 1995] and Freebase [Bollacker *et al.*, 2008]. FB15K [Bordes *et al.*, 2013] is extracted from a typical large-scale knowledge graph Freebase, and WN18 is extracted from the Wordnet. For a fair comparison, we did the same processing of the datasets as DKRL [Xie *et al.*, 2016] and SSP [Xiao *et al.*, 2017].

4.2 Knowledge Graph Link Prediction

Link prediction used in TransE [Bordes *et al.*, 2013] is a traditional evaluation method for knowledge graph completion.

Evaluation Protocol

First, for each testing triple (h, r, t), we replaced the tail t (or the head h) with every entity e in the knowledge graph, while the relation r was replaced in predicting relation. Then, a probabilistic score of this corrupted triple was calculated with the score function $f_{JNCTD}(h, r, t)$. By ranking these scores in ascending order, we then got the rank of the original triple. The evaluation metrics were the average of the ranks as Mean Rank and the proportion of testing triple whose rank is not larger than 10 as HITS@10. This is called "Raw" setting. Notice that if a corrupted triplet exists in the knowledge graph, as it is also correct, ranking it before the original triplet is not wrong. To eliminate this factor, we removed those corrupted triplets which exist in either training, valid, or testing set before getting the rank of each testing triplet. This setting is called "Filter" setting. In both settings, a higher HITS@10 and a lower Mean Rank mean better performance.

Table 1: Mean Rank and HITS@10 of link prediction of entities on FB15K and WN18.

FB15K	**Mean Rank**		**HITS@10**	
	Raw	Filter	Raw	Filter
TransE	210	119	48.5	66.1
TransH	212	87	45.7	64.4
DKRL(BOW)	200	113	44.3	57.6
DKRL(ALL)	181	91	49.6	67.4
JNCTD(Std.)	154	77	57.1	78.6
SSP(Joint)	163	82	57.2	79.0
JNCTD(WC)	**122**	**53**	54.9	76.1
JNCTD(PC)	127	55	**58.7**	**80.1**
WN18	**Mean Rank**		**HITS@10**	
	Raw	Filter	Raw	Filter
TransE	263	251	75.4	89.2
TransH	401	338	73	82.3
SSP(Std.)	204	193	**81.3**	91.4
SSP(Joint)	168	156	81.2	93.2
JNCTD(WC)	**157**	**144**	79.6	92.8
JNCTD(PC)	195	182	80.4	**93.9**

Parameter Settings

As the datasets are the same, we directly reprinted the experimental results of several baselines from the literature. We have attempted several settings on the validation dataset to get the best configuration. Under the "bern." sampling strategy, the optimal configurations of our model JNCTD are as follows. For FB15K, embedding dimension d = 100, learning rate α = 0.001, margin γ= 1.8, balance factor λ = 0.2, hyper-parameter k = 2 and a = 400, and μ = 0.1. For WN18, embedding dimension d = 100, learning rate α= 0.001, margin γ = 8.8, hyper-parameter k =2 and a = 100, balance factor λ = 0.2. We trained the model until convergence.

Table 2: Mean Rank and HITS@10 of knowledge graph completion (for predicting relations) on FB15K.

FB15K	**Mean Rank**		**HITS@10**	
	Raw	Filter	Raw	Filter
TransE	2.91	2.53	60.3	72.5
DKRL(BOW)	2.85	2.51	65.3	82.7
DKRL(ALL)	2.41	2.03	69.8	90.8
SSP(Std.)	**1.58**	**1.22**	69.9	89.2
SSP(Joint)	1.87	1.47	70.0	90.9
JNCTD(WC)	1.84	1.46	69.1	89.8
JNCTD(PC)	1.87	1.46	**70.5**	**91.2**

Results of Link Prediction and Relation Prediction

From the results of link prediction in Table 1 and relation prediction in Table 2, we observe that:

1. In link prediction, JNCTD outperforms all the baselines in all the tasks. The details are as follows: JNCTD(WC) improves 32 in Mean Rank and JNCTD(PC) increases 1.1% in Hits@10 against SSP in the dataset FB15K. JNCTD(WC) improves 11 in Mean Rank and JNCTD(PC) increases 0.7% in Hits@10 against SSP in dataset WN18. These improvements prove the effectivenesses of neighbor context and our model.
2. JNCTD(WC) gets the best Mean Rank. It shows the correctness of our theoretical analysis that the weighted composition method can collect more comprehensive semantic connections between entities. But on Hits@10 it has a slight decline. This is because the textual description and neighbor context are various and heterogeneous. Although we aligned them by conducting the two topic models and embedding model simultaneously, they still cause some noise. In FB15K, we checked the result and find that 4.1% triples which are Hits@10 in SSP, rank larger than 10 but less than 20 in our model. In fact, these ranking rises are slight and acceptable.
3. The fact that JNCTD(PC) has the best Hits@10 result also accords with our analysis. The projection way helps to find more precise semantic connections between entities from neighbor contexts and textual descriptions.
4. Figure 4 shows that the Mean Rank is along with the number of entities neighbors in the model JNCTD(WC). Those entities with larger counts of neighbors have better Mean Rank. This also demonstrates that neighbor context is effective. Furthermore, this trend verifies our view that entities connected with many entities in knowledge graph depend more on their neighbor contexts.

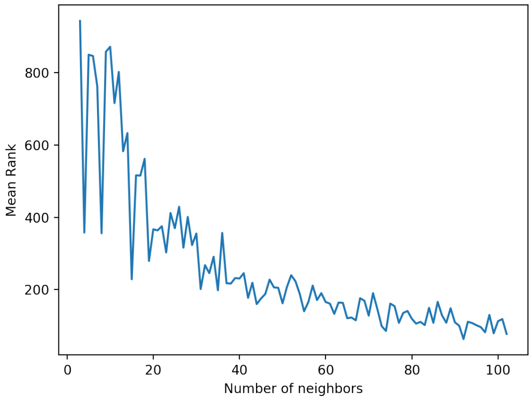


Figure 4: The relationship between the number of neighbors and Mean Rank

4.3 Entity Classification

Entity classification is a multilabel classification task aiming at predicting entity types, which is crucial and widely used in many NLP tasks [Neelakantan and Chang, 2015]. Almost every entity has types in Freebase, for example, the entity types of "*Roy Wood*" are *music producer*, *guitarist*, *actor*.

Evaluation Protocol

We adopted the same dataset as DKRL. It has 4,054 types in FB15K extracted from Freebase. We ranked those types by their frequency and selected top 50 types for classification (remove the type of common/topic which almost all entities have). The top 50 types cover 13,445 entities. We randomly split them into training set and test set, the training set has 12,113 entities while the test set has 1,332 entities.

In the training, we used the concatenation of neighbor context semantic vector $\mathbf{s_n}$, textual description semantic vector $\mathbf{s_d}$, and embedding vector $\mathbf{s_e}$ $(\mathbf{s_n}, \mathbf{s_d}, \mathbf{s_e})$ as entity representation, which are the features for the front-end classifier. For a fair comparison, our front-end classifier is also the Logistic Regression as DKRL and SSP in a one-versus-rest setting for multi-label classification. The evaluation is following [Neelakantan and Chang, 2015], which applies the mean average precision (MAP) that is commonly used in multi-label classification. Specifically, for an entity, if the methods predict a rank list of types and there are three correct types that lie in #1, #2, #4, the MAP is calculated as $\frac{1/1+2/2+3/4}{3}$.

Parameter Setting

As the datasets are the same, we directly reported the experimental results of several baselines. We have attempted several settings on the validation dataset to get the best configuration. Under the "bern." sampling strategy. The optimal configuration of our model JNCTD(WC) and JNCTD(PC) are as follows. For FB15K,embedding dimension $d = 100$,learning rate $\alpha = 0.001$, margin $\gamma = 1.8$ balance factor $\lambda = 0.2$, hyper-parameter k = 2 and a =400, and $\mu = 0.1$.

Results of Entity Classification

From Table 3 we observed that: JNCTD outperforms all other models in FB15K. This result justifies the effectiveness of neighbor context. With more features, we certainly have a greater grasp of inferring the entity types. This result proves the correctness of our use of topic model to extract features from neighbor context. Other models could be treated as missing neighbor contexts information.

Table 3: The MAP result of Entity Classification.

Metrics	**FB15K**
TransE	87.8
BOW	86.3
DKRL(BOW)	89.3
DKRL(ALL)	90.1
NMF	86.1
SSP(Std.)	93.2
SSP(Joint)	94.4
JNCTD(WC)	96.4
JNCTD(PC)	**96.5**

5 Case Study

In this section, we analyze our model in two cases. One is to show two examples in detail, and the other is to make a statistic analysis of the result of link prediction.

5.1 Example Case

For the example cases, the triple $(Boston\ Red\ Sox, /sport\_team/draft\_picks/school, St.\ John's\ University)$ ranks 205 in SSP while 13 in JNCTD(WC) and 26 JNCTD(PC). We observed the details of its neighbors and found that both of their neighbors share relevant topics of "university" and "sports". Another triple $(Murder\ on\ the\ Orient\ Express, /film/subject,\ train)$ ranks 937 in SSP, which means an impossible fact. After introducing neighbor context, it ranks 107 in JNCTD(WC) and 116 in JNCTD(PC), which is more plausible. Therefore, there are two benefits of introducing neighbor context. One is we can correctly classify and discriminate some triples which are difficult to identify correctness. Another is when a correct triple makes much loss, after our score function it could be relatively smaller.

5.2 Statistic Analysis

We also made a statistic analysis of the result in link prediction. As reported in Table 4, the number in each cell means the number of triples whose rank are larger than m in SSP and less than n in our models. For instance, the number 401 means there are 401 triples whose ranks are less than 100 in JNCTD(WC), while more than 300 in SSP. The results show that our model is to improve SSP with neighbor context.

Table 4: Link Rrediction rank statistics of JNCTD. The number in each cell indicates the number of triples.

	JNCTD(WC) $\leq$ 100	JNCTD(PC)$\leq$ 100
SSP $\geq$ 1000	29	16
SSP $\geq$ 500	175	96
SSP $\geq$ 300	401	231

6 Conclusion and Future Work

We have proposed a novel text-aware model named Joint Neighbor Contexts and Textual Descriptions (JNCTD) for representation learning of knowledge graph with entity neighbor context and textual description. We have introduced topic model to embed the neighbor context and applied weighted composition and projection composition methods to combine the embeddings of neighbor contexts and textual descriptions. Two evaluation tasks including link prediction and entity classification on WN18 and FB15K show that our model consistently outperforms other baselines including DKRL and SSP.

We will explore the following research directions in future:

1. The JNCTD model only considers the neighbor context for representation learning, while there is some graph context information like relation context, path context. We may take advantages of that rich information in future.
2. Currently, in JNCTD(WC) only the number of neighbors effects the weight parameters. But we should also take the reliability of textual description into account, which could be helpful to learn a better semantic hyperplane.
3. We may extend our method of processing neighbor context to more models, for example, PTransE [Lin *et al.*, 2015a] needs to design a path between entities, we can take neighbor context into consideration.

References

[Blei, 2012] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase:a collaboratively created graph database for structuring human knowledge. In *Proceedings of 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 *Proceedings of NIPS*, pages 2787–2795, 2013.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research*, 9:249–256, 2010.

[He *et al.*, 2015] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of CIKM*, pages 623–632, 2015.

[Ji *et al.*, 2015] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL*, pages 687–696, 2015.

[Ji *et al.*, 2016] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of AAAI*, pages 985–991, 2016.

[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 *Proceedings of 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 *Proceedings of AAAI*, pages 2181–2187, 2015.

[Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[Neelakantan and Chang, 2015] Arvind Neelakantan and Ming Wei Chang. Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of HLT-NAACL*, pages 515–525, 2015.

[Shi *et al.*, 2017] Jun Shi, Huan Gao, Guilin Qi, and Zhangquan Zhou. Knowledge graph embedding with triple context. In *Proceedings of CIKM*, pages 2299–2302, 2017.

[Stevens *et al.*, 2012] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of EMNLP-CoNLL*, pages 952–961, 2012.

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

[Xiao *et al.*, 2016] Han Xiao, Minlie Huang, and Xiaoyan Zhu. A generative model for knowledge graph embedding. In *Proceedings of ACL*, pages 2316–2325, 2016.

[Xiao *et al.*, 2017] Han Xiao, Minlie Huang, Lian Meng, and Xiaoyan Zhu. Sssp: Semantic space projection for knowledge graph embedding with text descriptions. In *Proceedings of AAAI*, pages 3104–3110, 2017.

[Xie *et al.*, 2016] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of AAAI*, pages 2659–2665, 2016.

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