

SSP: Semantic Space Projection for Knowledge Graph Embedding with Text Descriptions

Han Xiao,* Minlie Huang,* Lian Meng, Xiaoyan Zhu

State Key Lab. of Intelligent Technology and Systems,
National Lab. for Information Science and Technology,

Dept. of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China
bookman@vip.163.com; {aihuang,zxy-dcs}@tsinghua.edu.cn; mengl15@foxmail.com;

*Correspondence Authors: <http://www.ibookman.net>, <http://www.aihuang.org>

Abstract

Knowledge graph embedding represents entities and relations in knowledge graph as low-dimensional, continuous vectors, and thus enables knowledge graph compatible with machine learning models. Though there have been a variety of models for knowledge graph embedding, most methods merely concentrate on the fact triples, while supplementary textual descriptions of entities and relations have not been fully employed. To this end, this paper proposes the *semantic space projection (SSP)* model which jointly learns from the symbolic triples and textual descriptions. Our model builds interaction between the two information sources, and employs textual descriptions to discover semantic relevance and offer precise semantic embedding. Extensive experiments show that our method achieves substantial improvements against baselines on the tasks of knowledge graph completion and entity classification. *Papers, Posters, Slides, Datasets and Codes: <http://www.ibookman.net/conference.html>*

Introduction

Knowledge graph provides an effective basis for NLP in many tasks such as question answering, web search and semantic analysis. In order to provide a numerical computation framework for knowledge graph, knowledge graph embedding projects the entities and relations to a continuous low-dimensional vector space. More specifically, a fact in knowledge graph is usually represented as a symbolic triple (h, r, t) , while knowledge graph embedding attempts to represent the symbols with vectors, say \mathbf{h} , \mathbf{r} , \mathbf{t} . To this end, a number of embedding methods have been proposed, such as TransE (Bordes et al. 2013), PTransE (Lin, Liu, and Sun 2015), KG2E (He et al. 2015), etc.

As a key branch of embedding models, the translation-based methods adopt the principle of translating the head entity to the tail one by a relation-specific vector, or formally as $\mathbf{h} + \mathbf{r} = \mathbf{t}$. As Fig.1 shows, in the knowledge graph, the entities such as h , t have textual descriptions, which contain much supplementary semantic information about knowledge triples.

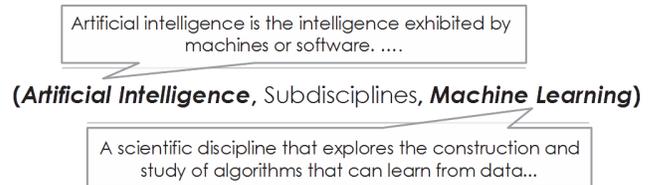


Figure 1: Textual descriptions for entities in a fact triple.

Despite the success of conventional knowledge graph embedding models, there are still two reasons why textual descriptions would be necessary in this task: *discovering semantic relevance* and *offering precise semantic expression*.

Firstly, the semantic relevance between entities is capable to recognize the true triples, which are difficult to be inferred only with fact triples. For example, the triple $(Anna\ Roosevelt, Parents, Franklin\ Roosevelt)$, indicates “Franklin Roosevelt” is the parent of “Anna Roosevelt”. However, it’s quite difficult to infer this fact merely from other symbolic triples. In contrast, in the textual description of the head entity, there are many keywords such as “Roosevelt” and “Daughter of the President”, which may infer the fact triple easily. Specifically, we measure the possibility of a triple by projecting the loss onto a hyperplane that represents the semantic relevance between entities. Thus, it is always possible to accept a fact triple so long as the l_2 -norm of the projected loss vector onto the semantic hyperplane is sufficiently small.

Secondly, precise semantic expression could promote the discriminative ability between two triples. For instance, when we query about the profession of “Daniel Sturgeon”, there are two possible candidates: “politician” and “lawyer”. It’s hard to distinguish if only focusing on the symbolic triples. However, the textual description of “Daniel Sturgeon” is full of politics-related keywords such as “Democratic Party”, “State Legislature” etc. and even “Politician”. The textual descriptions help to refine the topic of “Daniel Sturgeon” in a more precise way from the social celebrities to the government officers, which makes the true answer “politician” more preferable. Formally, even though the loss vectors of the two facts are almost of equal norm, after respectively projected onto the “politician” and “lawyer” related semantic hyperplanes, the losses

are distinguished reasonably. In this way, precise semantic expression refines the embedding.

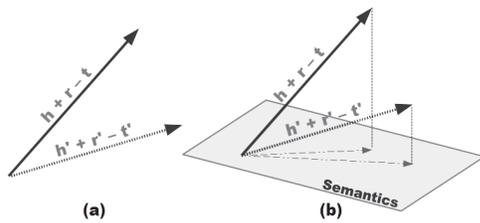


Figure 2: Simple illustration of TransE and SSP where $\mathbf{h} + \mathbf{r} - \mathbf{t}$ is the loss vector. The loss vectors of the two triples in (a) are length-equal, thus, it is hard to identify the correctness. In (b), we introduce a semantic hyperplane, and project the loss vectors to the hyperplane to consider the semantics of triples.

The existing embedding methods with textual semantics such as DKRL (Xie et al. 2016) and “Jointly” (Zhong et al. 2015), have achieved much success. But there is still an issue to be addressed, the *weak-correlation modeling* issue that current models could hardly characterize the strong correlations between texts and triples. In DKRL, for a triple, the embedding vector of the head entity is translated to that of the tail one as well as possible, and then the vectors of texts and entities are concatenated as the final embedding vectors. While in the “Jointly” model the authors attempt to generate coherent embeddings of the corresponding entity and text. Both DKRL and “Jointly” apply first-order constraints which are weak in capturing the correlation of texts and triples. ***It’s noteworthy that triple embedding is always the main procedure and textual descriptions must interact with triples for better embedding.*** Only in this way, the semantic effects could make more senses. For the above example of “Daniel Sturgeon”, the textual descriptions imply two candidate answers “Banker” and “Politician”. Thus, by considering both triples and texts, we can obtain the true answer. Therefore, we focus on the stronger semantic interaction by projecting triple embedding onto a semantic subspace such as hyperplane, as shown in Fig.2. Mathematically, the quadratic constraint is adopted to model the strong correlation, thus the embedding topologies are sufficiently semantics-specific.

We evaluate the effectiveness of our model *Semantic Subspace Projection (SSP)* with two tasks on three benchmark datasets that are the subsets of Wordnet (Miller 1995) and Freebase (Bollacker et al. 2008). Experimental results on the datasets show that our model consistently outperforms the other baselines with remarkable improvement.

Contributions. We propose a knowledge graph embedding method SSP which models the strong correlations between the symbolic triples and the textual descriptions by performing the embedding process in a semantic subspace. Besides, our method outperforms all the baselines on the tasks of knowledge graph completion and entity classification, which justifies the effectiveness of our proposal.

Related Work

We have surveyed the previous studies and categorized the embedding methods into two branches: *Triple-only Embedding* models that use only symbolic triples and “*Text-Aware*” *Embedding* models that employ textual descriptions.

Triple-only Embedding Models

TransE (Bordes et al. 2013) is a pioneering work for this branch, which translates the head entity to the tail one by the relation vector, or $\mathbf{h} + \mathbf{r} = \mathbf{t}$. Naturally, the L_2 norm of the loss vector is the score function, which measures the plausibility of a triple and a smaller score is better.

The following variants transform entities into different subspaces. ManifoldE (Xiao, Huang, and Zhu 2016a) opens a classic branch, where a manifold is applied for the translation. TransH (Wang et al. 2014b) utilizes the relation-specific hyperplane to lay the entities. TransR (Lin et al. 2015) applies the relation-related matrix to rotate the embedding space. Similar researches include TransG (Xiao, Huang, and Zhu 2016b), TransD (Ji et al.), (Feng et al. 2016) and TransM (Fan et al. 2014).

Further researches encode additional structural information into embedding. PTransE (Lin, Liu, and Sun 2015) is a path-based model, simultaneously considering the information and confidence level of a path in the knowledge graph. (Wang, Wang, and Guo 2015) incorporate the rules to restrict the embeddings for the complex relation types such as 1-N, N-1 and N-N. SSE (Guo et al. 2015) aims at discovering the geometric structure of embedding topologies and then based on these assumptions, designs a semantically smoothing score function. Also, KG2E (He et al. 2015) involves probabilistic analysis to characterize the uncertainty concepts of knowledge graph. There are also some other works such as SE (Bordes et al. 2011), LFM (Jenatton et al. 2012), NTN (Socher et al. 2013) and RESCAL (Nickel, Tresp, and Krieger 2011), TransA (Xiao et al. 2015), etc.

Text-aware Embedding Models

“Text-Aware” Embedding, which attempts to representing knowledge graph with textual information, generally dates back to NTN (Socher et al. 2013). NTN makes use of entity name and embeds an entity as the average word embedding vectors of the name. (Wang et al. 2014a) attempts to aligning the knowledge graph with the corpus then jointly conducting knowledge embedding and word embedding. However, the necessity of the alignment information limits this method both in performance and practical applicability. Thus, (Zhong et al. 2015) proposes the “Jointly” method that only aligns the freebase entity to the corresponding wiki-page. DKRL (Xie et al. 2016) extends the translation-based embedding methods from the triple-specific one to the “Text-Aware” model. More importantly, DKRL adopts a CNN-structure to represent words, which promotes the expressive ability of word semantics. Generally speaking, by jointly modeling knowledge and texts, *text-aware embedding models* obtains the state-of-the-art performance.

Methodology

In this section, we first introduce our model and then provide two different perspective to address the advantages of our model. First of all, let us introduce some notations: all the symbols h, t indicate the head and tail entity, respectively. \mathbf{h} (or \mathbf{t}) is the embedding of the entity from the triples, \mathbf{s}_h (or \mathbf{s}_t) is the semantic vector generated from the texts, and d is the dimension of embedding.

The data involved in our model are the knowledge triples and the textual descriptions of entities. In experiments, we adopt the “entity descriptions” of Freebase and the textual definitions of Wordnet as textual information.

Model Description

Previous analysis in the introduction suggests to characterize the strong correlation between triples and texts. For the purpose of interacting between the symbolic triples and textual descriptions, this paper attempts to restrict the embedding procedure of a specific triple in the semantic subspace. Specifically, we leverage a hyperplane with normal vector $\mathbf{s} \doteq \mathcal{S}(\mathbf{s}_h, \mathbf{s}_t)$ as the subspace, where $\mathcal{S} : \mathbb{R}^{2d} \mapsto \mathbb{R}^d$ is the semantic composition function which will be discussed in the next section, and $\mathbf{s}_h, \mathbf{s}_t$ is the head-specific and tail-specific semantic vectors, respectively.

The score function in the translation-based methods is $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$, which means the triple embedding focuses on the loss vector $\mathbf{e} \doteq \mathbf{h} + \mathbf{r} - \mathbf{t}$. According to our motivation, assuming \mathbf{e} is length-fixed, the target is to maximize the component inside the hyperplane, which is $\|\mathbf{e} - \mathbf{s}^\top \mathbf{e}\|_2^2$. In detail, the component of the loss in the normal vector direction is $(\mathbf{s}^\top \mathbf{e})$, then the other orthogonal one, that is inside the hyperplane, is $(\mathbf{e} - \mathbf{s}^\top \mathbf{e})$.

It’s natural that the norm of the loss vector should also be constrained. To this end, we introduce a factor λ to balance the two parts, formally as:

$$f_r(h, t) = -\lambda \|\mathbf{e} - \mathbf{s}^\top \mathbf{e}\|_2^2 + \|\mathbf{e}\|_2^2$$

where λ is a suitable hyper-parameter. Moreover, a smaller score means the triple is more plausible. For clarity, the definitions of the symbols are boxed. *Notably, the projection part in our score function is negative, so smaller value means less loss.*

Semantic Vector Generation

There are at least two methods that could be leveraged to generate the semantic vectors: topic model (Blei 2012) such as LSA, LDA, NMF (Stevens et al. 2012) and word embedding such as CBOW (Mikolov, Yih, and Zweig 2013), Skip-Gram (Mikolov et al. 2013). More concretely, this paper adopts the topic model, treating each entity description as a document and then obtains the topic distribution of document as the semantic vector of entity. The entities are usually organized by the topic in knowledge base, for example, “entity type” is used in Freebase to categorize entities. Therefore, we conjecture that the topic model could be more suitable. Notably, the word embedding would also work well though maybe not better.

Given the pre-trained semantic vectors, our model fixes them and then optimizes the other parameters. We call this setting **Standard** (short as **Std.**). The reason why we could not adapt all the parameters, is that the training procedure would refill the semantic vectors and flush the semantics out. For the purpose of jointly learning the semantics and the embeddings, we conduct the topic model and the embedding model, simultaneously. In this way, the symbolic triples also impose a positive effect on the textual semantics and we call this setting **Joint**.

As each component of a semantic vector indicates the relevant level to a topic, we suggest the semantic composition should take the addition form:

$$\mathcal{S}(\mathbf{s}_h, \mathbf{s}_t) = \frac{\mathbf{s}_h + \mathbf{s}_t}{\|\mathbf{s}_h + \mathbf{s}_t\|_2^2}$$

where the normalization is applied to make a normal vector. Since the largest components represent the topics, the addition operator corresponds to the union of topics, making the composition indicate the entire semantics. For example, when $\mathbf{s}_h = (0.1, 0.9, 0.0)$ and $\mathbf{s}_t = (0.8, 0.0, 0.2)$, the topic of the head entity is #2 and that of the tail is #1, while the composition is $\mathbf{s} = (0.45, 0.45, 0.10)$, corresponding to the topic of #1, #2, which is accordant to our intuition.

Correlation Perspective

Specifically, our model attempts to lay the loss $\mathbf{h}' - \mathbf{t}$ onto the hyperplane, where $\mathbf{h}' \doteq \mathbf{h} + \mathbf{r}$ is the translated head entity. Mathematically, if a line lies on a hyperplane, so do all the points of this line. Correspondingly, the loss lays on the hyperplane, implying the head and tail entity also lay on it, as the beginning and ending point. ***Thus, there exists the important restriction, that the entities co-occur in a triple should be embedded in the semantic space composed by the associated textual semantics.*** This restriction is implemented as a quadric form to characterize the strong correlation between texts and triples, in other words, to interact with both the information sources. A strong interaction between the textual descriptions and symbolic triples complements each other in a more semantics-specific form, which guarantees the semantic effects. More concretely, the embeddings are decided in the training procedure not only by triples but also by textual semantics, based on which, our embedding topologies are semantically different from the other methods.

Semantic Perspective

There are two semantic effects for textual descriptions: discovering semantic relevance and offering precise semantic expression. ***Our model characterizes the strong correlations with a semantic hyperplane, which is capable of taking the advantages of both two semantic effects.***

Firstly, according to the correlation perspective, the entities which are semantically relevant, approximately lay on a consistent hyperplane. Therefore, the loss vector between them ($\mathbf{h}' - \mathbf{t}$) is also around the hyperplane. Based on this geometric insight, when a head entity matches a negative tail, the triple is far from the hyperplane, making a large loss to be classified. Conversely, even if a correct triple makes

much loss, the score function after projected onto the hyper-plane could be relatively smaller (or better). By this mean, the semantic relevance achieved from the texts, promotes embedding. For instance, the fact triple (*Portsmouth Football Club, Locate, Portsmouth*) could hardly be inferred only within the triple embedding. It ranks 11,549 out of 14,951 by TransE in link prediction, which means a totally impossible fact. But the keywords “Portsmouth”, “England”, and “Football” occur many times in both the textual descriptions, making the two entities semantically relevant. Unsurprisingly, after the semantic projection, the case ranks 65 out of 14,951 in our model, which is much more plausible.

Secondly, all the equal-length loss vectors¹ in TransE are equivalent in term of discrimination since the score function of TransE is $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$, which leads to the weak distinction. However, with textual semantics, the discriminative ability could be strengthened in our model. Specifically, the equal-length loss vectors are measured with the projection onto the corresponding semantic hyperplanes, which makes a reasonable division of the losses. For an instance of the query about which film “*John Powell*” contributes to, there are two candidate entities, that the true answer “*Kung Fu Panda*” and the negative one “*Terminator Salvation*”. Without textual semantics, it’s difficult to discriminate, thus the losses calculated by TransE are 8.1 and 8.0, respectively, leading to a hard decision. Diving into the textual semantics, we discover, “*John Powell*” is much relevant to the topic of “*Animated Films*”, which matches that of “*Kung Fu Panda*” and does not for the other. Based on this fact, both the query and the true answer lie in the “*Animated Films*”-directed hyperplane, whereas the query and the negative one do not co-occur in the corresponding associated semantic hyperplane. Thus, the projected loss of the true answer could be much less than that of the false one. Concretely, the losses in our model are 8.5 and 10.8, respectively, which are sufficient for discrimination.

Objectives & Training

There are two parts in the objective function, which are respectively embedding-specific and topic-specific. To balance the two parts, a hyper-parameter μ is introduced. Overall, the total loss is:

$$\mathcal{L} = \mathcal{L}_{embed} + \mu\mathcal{L}_{topic} \quad (1)$$

Notably, there is only the first part in the **Standard** setting where $\mu = 0$ in fact.

In term of the embedding-related objective, the rank-based hinge loss is applied, which means to maximize the discriminative margin between the golden triples and the negative ones:

$$\mathcal{L}_{embed} = \sum_{\substack{(h, r, t) \in \Delta \\ (h', r', t') \in \Delta'}} [f_{r'}(h', t') - f_r(h, t) + \gamma]_+$$

where Δ is the set of golden triples and Δ' is that of the negative ones. γ is the margin, and $[\cdot]_+ = \max(\cdot, 0)$ is the hinge loss. The false triples are sampled with “Bernoulli

¹Two vectors have equal-length iff they have the same l_2 norm.

Sampling Method” as introduced in (Wang et al. 2014b) and the method selects the negative samples from the set of

$$\begin{aligned} \{(h', r, t) | h' \in E\} \cup \{(h, r, t') | t' \in E\} \\ \cup \{(h, r', t) | r' \in R\} \end{aligned}$$

We initialize the embedding vectors by the similar methods used in the deep neural network (Glorot and Bengio 2010) and pre-train the topic model with Non-negative Matrix Factorization (NMF) (Stevens et al. 2012). The stochastic gradient descent algorithm (SGD) is adopted in the optimization.

For the topic-related objective, we take the advantage of the NMF Topic Model (Stevens et al. 2012), which is both simple and effective. Then we re-write the target as an inner-product form with the L_2 -loss, stated as:

$$\mathcal{L}_{topic} = \sum_{e \in E, w \in D_e} (C_{e,w} - \mathbf{s}_e^\top \mathbf{w})^2 \quad (2)$$

$$\mathbf{s}_e \geq \mathbf{0}, \mathbf{w} \geq \mathbf{0} \quad (3)$$

where E is the set of entities, and D_e is the set of words in the description of entity e . $C_{e,w}$ is the times of the word w occurring in the description e . \mathbf{s}_e is the semantic vector of entity e and \mathbf{w} is the topic distribution of word w . Similarly, SGD is applied in the optimization.

Theoretically, our computation complexity is comparable to TransE, as $\mathcal{O}(\nu \times \mathcal{O}(TransE))$, and the small constant ν is caused by the projection operation and topic calculation. In practice, TransE costs 0.28s for one round in Link Prediction and our model costs 0.36s in the same setting. In general, TransE is most efficient among all the translation-based methods, while our method could be comparable to TransE in running time, justifying the efficiency of our model.

Experiments

Datasets & General Settings

Our experiments are conducted on three public benchmark datasets that are the subsets of Wordnet and Freebase. About the statistics of these datasets, we strongly suggest the readers to refer to (Xie et al. 2016) and (Lin et al. 2015). The entity descriptions of FB15K and FB20K are the same as DKRL (Xie et al. 2016), each of which is a small part of the corresponding wiki-page. The textual information of WN18 is the definitions that we extract from the Wordnet. Notably, for the zero-shot learning, FB20K is involved, which is also built by the authors of DKRL.

Knowledge Graph Completion

Evaluation Protocol. The same protocol used in previous studies, is adopted. First, for each testing triple (h, r, t) , we replace the tail t (or the head h) with every entity e in the knowledge graph. Then, a probabilistic score of this corrupted triple is calculated with the score function $f_r(h, t)$. By ranking these scores in ascending order, we then get the rank of the original triple. The evaluation metrics are 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. When we filter out the corrupted triples that exist in the training, validation, or test

datasets, this is the “Filter” setting. If a corrupted triple exists in the knowledge graph, ranking it ahead the original triple is also correct. To eliminate this effect, the “Filter” setting is more preferred. In both settings, a higher HITS@10 and a lower Mean Rank mean better performance.

Table 1: Mean Rank and HITS@10 of Knowledge Graph Completion (For Predicting Entity) on FB15K and WN18.

FB15K	Mean Rank		HITS@10	
TransE	210	119	48.5	66.1
TransH	212	87	45.7	64.4
Jointly	167 ¹	39 ¹	51.7 ¹	77.3 ¹
DKRL(BOW)	200	113	44.3	57.6
DKRL(ALL)	181	91	49.6	67.4
SSP (Std.)	154	77	57.1	78.6
SSP (Joint)	163	82	57.2	79.0
WN18	Mean Rank		HITS@10	
TransE	263	251	75.4	89.2
TransH	401	338	73.0	82.3
SSP (Std.)	204	193	81.3	91.4
SSP (Joint)	168	156	81.2	93.2

Implementation. As the datasets are the same, we directly reprint 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 SSP are as follows. For WN18, embedding dimension $d = 100$, learning rate $\alpha = 0.001$, margin $\gamma = 6.0$, balance factor $\lambda = 0.2$ and for SSP(Joint) $\mu = 0.1$. For FB15K, embedding dimension $d = 100$, learning rate $\alpha = 0.001$, margin $\gamma = 1.8$, balance factor $\lambda = 0.2$ and for SSP(Joint) $\mu = 0.1$. We train the model until convergence.

Table 2: Mean Rank and HITS@10 of Knowledge Graph Completion (For Predicting Relation) on FB15K.

FB15K	Mean Rank		HITS@10	
TransE	2.91	2.53	69.5	90.2
TransH	8.25	7.91	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

Results Evaluation results are reported in Tab.1 and Tab.2. Note that “Jointly” refers to (Zhong et al. 2015). We observe that:

1. SSP outperforms all the baselines in all the tasks, demonstrating the effectiveness of our models and the correctness of our theoretical analysis. Specifically, SSP(Joint)

¹This method involves much more extra text corpus, thus it’s unfair to directly compare with others.

Table 3: The MAP result of Entity Classification

Metrics	FB15K	FB20K
TransE	87.8	-
BOW	86.3	57.5
DKRL(BOW)	89.3	52.0
DKRL(ALL)	90.1	61.9
NMF	86.1	59.6
SSP (Std.)	93.2	-
SSP (Joint)	94.4	67.4

improves much more than SSP(Std.) for jointly learning the textual semantics and symbolic triples.

2. DKRL and “Jointly” model only consider the first-order constraints, which interact between the textual and symbolic information, unsatisfactorily. By focusing on the strong correlation, SSP outperforms them. Notice that the “Jointly” model involves much more additional data to produce the result, but SSP also has an remarkable advantage against it. Though TransH is also a hyperplane-based method, SSP adopts the hyperplane in a semantics-specific way rather than a simple relation-specific form.
3. TransE could be treated as missing textual descriptions, and DKRL(BOW) could be viewed as missing symbolic triples. SSP (Joint) improves 12.4% against TransE while 20.9% against DKRL(BOW), illustrating the triple embedding is always the key point and the interactions between the two information sources is a key factor.

Entity Classification

This task is essentially a multi-label classification, focusing on predicting entity types, which is crucial and widely used in many NLP & IR tasks (Neelakantan and Chang 2015). The entity in Freebase always has types, for instance, the entity types of “Scots” are *Human Language*, *Rosetta Language*.

We adopt the same datasets as DKRL, for the details of which, we refer readers to (Xie et al. 2016). Overall, this is a multi-label classification task with 50 classes, which means for each entity, the method should provide a set of types rather than a single type.

Evaluation Protocol In the training, we use the concatenation of semantic vector and embedding vector (s_e, e) as entity representation, which is the feature for the front-end classifier. For a fair comparison, our front-end classifier is also the Logistic Regression as DKRL 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 $\frac{1/1+2/2+3/4}{3}$. For FB20K, the methods could only make use of the descriptions. Obviously, FB20K is a zero-shot scenario which is really hard.

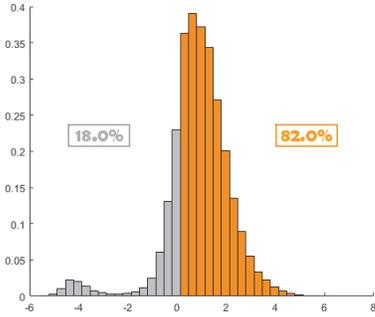


Figure 3: Precise semantic expression analysis for SSP (Std.). The x-axis indicates the score difference, where a bigger value means better. The y-axis means the proportion of the corresponding triple pairs. The gray part indicates that both the TransE and SSP make mistakes while the yellow colored part means that SSP succeeds but TransE fails. The statistics are similar for the SSP(J.) setting.

Implementation. As the datasets are the same, we directly report 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 SSP are as follows. For FB15K, embedding dimension $d = 100$, learning rate $\alpha = 0.001$, margin $\gamma = 1.8$, balance factor $\lambda = 0.2$ and for SSP(Joint) $\mu = 0.1$. For FB20K, embedding dimension $d = 100$, learning rate $\alpha = 0.001$, margin $\gamma = 1.8$, balance factor $\lambda = 0.2$ and for SSP(Joint) $\mu = 0.1$. We train the model until convergence.

Results The evaluation results are listed in Tab. 3. Notably, TransE is only triple-specific and SSP (Std.) performs as well as the NMF in the zero-shot learning. Thus they are trivial for FB20K. The observations are as follows:

1. Overall, our method SSP yields the best accuracy, justifying the effectiveness of our method.
2. Compared to NMF and TransE, the results show that the interactions between triples and texts make a key difference and only one of them could hardly produce an effective performance. The promotion of SSP (Joint) in FB20K demonstrates the triple embedding also has a positive effect on textual semantics.
3. Compared to TransE, the improvements illustrate the effectiveness of semantic integration. Compared to DKRL, the promotion demonstrates the important role of modeling strong correlation.

Semantic Relevance Analysis

One benefit of modeling semantic relevance is the ability to correctly classify the triples that could not be discriminated by just using the information from symbolic triples. Hence, we make statistic analysis of the results in Link Prediction, as reported in the Tab.4. The number in each cell means the number of triples whose rank is larger than m in TransE and less than n in our models. For instance, the number 601

means there are 601 triples whose ranks are less than 100 in SSP(S.) while their ranks are more than 500 in TransE. Note that SSP(S.) indicates the standard setting, and SSP(J.) means the joint setting. The statistic results indicate that many triples benefit from the semantic relevance offered by textual descriptions. The experiments also justify the theoretical analysis about the semantic relevance and demonstrate the effectiveness of our models.

We also study some specific cases. The triple (*Luis Miguel Gallego Basteri*, *Music Artist Genre*, *Ballad*) ranks at 1,602 in TransE while 8 in SSP. Looking into the textual details, we discover the two entities share the relevant topics of “*Cross-Country*” and “*Music*”. For another triple that (*Laura Elizabeth Dern*, *Film Performance*, *Alice Doesn’t Live Here Anymore*), it ranks at 1,137 in TransE while 77 in SSP, where the two entities share relevant semantics such as “*Best Actress*” and “*Academy Award*”.

Table 4: Rank statistics of Link Prediction. The number in each cell indicates the number of triples, and r is the rank given by the corresponding models.

	SSP(S.) _{#r≤100}	SSP(J.) _{#r≤100}
TransE _{#r≥500}	601	672
TransE _{#r≥1000}	275	298
TransE _{#r≥2000}	80	89
TransE _{#r≥3000}	32	39

Precise Semantic Expression Analysis

As discussed previously, precise semantic expression leads to better discrimination. To justify this claim, we have collected the negative triples by link prediction, which are scored slightly better than the golden ones by TransE (that means, these are hard examples for TransE), and then plotted the SSP score difference between each corresponding pair of the negative and golden triples as Fig.3 shows. All these triples are predicted wrongly by TransE, but with precise semantic expression, our model correctly distinguishes 82.0% (Std.) and 83.2% (Joint) of them. In the histogram, the right bars indicate that SSP makes correct decision while TransE fails and the left bars mean both SSP and TransE fail. The experiments justify the theoretical analysis about the precise semantic expression and demonstrate the effectiveness of our models.

Conclusion

In this paper, we propose the knowledge graph embedding model SSP, which jointly learns from the symbolic triples and textual descriptions. SSP could interact between the triples and texts by characterizing the strong correlations between fact triples and textual descriptions. The textual descriptions have much effect on discovering semantic relevance and then offering precise semantic expression. Extensive experiments show our method achieves the substantial improvements against the state-of-the-art baselines.

Acknowledgement. This work was partly supported by the National Basic Research Program (973 Program) under

grant No. 2013CB329403, and the National Science Foundation of China under grant No.61272227/61332007.

References

- Blei, D. M. 2012. Probabilistic topic models. *Communications of the ACM* 55(4):77–84.
- Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. ACM.
- Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y.; et al. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-fifth AAAI Conference on Artificial Intelligence*.
- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, 2787–2795.
- Fan, M.; Zhou, Q.; Chang, E.; and Zheng, T. F. 2014. Transition-based knowledge graph embedding with relational mapping properties. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, 328–337.
- Feng, J.; Huang, M.; Yang, Y.; and Zhu, X. 2016. GAKE: Graph aware knowledge embedding. In *COLING 2016, 26th International Conference on Computational Linguistics*.
- Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, 249–256.
- Guo, S.; Wang, Q.; Wang, B.; Wang, L.; and Guo, L. 2015. Semantically smooth knowledge graph embedding. In *Proceedings of ACL*.
- He, S.; Liu, K.; Ji, G.; and Zhao, J. 2015. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 623–632. ACM.
- Jenatton, R.; Roux, N. L.; Bordes, A.; and Obozinski, G. R. 2012. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems*, 3167–3175.
- Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. Knowledge graph embedding via dynamic mapping matrix.
- Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Lin, Y.; Liu, Z.; and Sun, M. 2015. Modeling relation paths for representation learning of knowledge bases. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.
- Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, 746–751.
- Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.
- Neelakantan, A., and Chang, M.-W. 2015. Inferring missing entity type instances for knowledge base completion: New dataset and methods. *arXiv preprint arXiv:1504.06658*.
- Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 809–816.
- Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, 926–934.
- Stevens, K.; Kegelmeyer, P.; Andrzejewski, D.; and Buttlar, D. 2012. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 952–961. Association for Computational Linguistics.
- Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014a. Knowledge graph and text jointly embedding. In *EMNLP*, 1591–1601. Citeseer.
- Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014b. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1112–1119.
- Wang, Q.; Wang, B.; and Guo, L. 2015. Knowledge base completion using embeddings and rules. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.
- Xiao, H.; Huang, M.; Hao, Y.; and Zhu, X. 2015. TransA: An adaptive approach for knowledge graph embedding. *arXiv preprint arXiv:1509.05490*.
- Xiao, H.; Huang, M.; and Zhu, X. 2016a. From one point to a manifold: Knowledge graph embedding for precise link prediction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.
- Xiao, H.; Huang, M.; and Zhu, X. 2016b. TransG : A generative model for knowledge graph embedding. In *Proceedings of the 29th international conference on computational linguistics*. Association for Computational Linguistics.
- Xie, R.; Liu, Z.; Jia, J.; Luan, H.; and Sun, M. 2016. Representation learning of knowledge graphs with entity descriptions.
- Zhong, H.; Zhang, J.; Wang, Z.; Wan, H.; and Chen, Z. 2015. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of EMNLP*, 267–272.