Knowledge Graph Representation with Jointly Structural and Textual Encoding

Jiacheng Xu[♠♣] **Xipeng Qiu**[*◇♣] **Kan Chen**[◇♣] **Xuanjing Huang**[◇♣]
[♣]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
[◇]School of Computer Science, Fudan University, Shanghai, China
[♠]Software School, Fudan University, Shanghai, China

Abstract

The objective of knowledge graph embedding is to encode both entities and relations of knowledge graphs into continuous low-dimensional vector spaces. Previously, most works focused on symbolic representation of knowledge graph with structure information, which can not handle new entities or entities with few facts well. In this paper, we propose a novel deep architecture to utilize both structural and textual information of entities. Specifically, we introduce three neural models to encode the valuable information from text description of entity, among which an attentive model can select related information as needed. Then, a gating mechanism is applied to integrate representations of structure and text into a unified architecture. Experiments show that our models outperform baseline and obtain state-of-the-art results on link prediction and triplet classification tasks.

1 Introduction

Knowledge graphs have been proved to benefit many artificial intelligence applications, such as relation extraction, question answering and so on. A knowledge graph consists of multi-relational data, having entities as nodes and relations as edges. An instance of fact is represented as a triplet (*Head Entity, Relation, Tail Entity*), where the *Relation* indicates a relationship between these two entities. In the past decades, great progress has been made in building large scale knowledge graphs, such as WordNet[Miller, 1995], Freebase [Bollacker *et al.*, 2008]. However, most of them have been built either collaboratively or semi-automatically and as a result, they often suffer from incompleteness and sparseness.

The knowledge graph completion is to predict relations between entities based on existing triplets in a knowledge graph. Recently, a new powerful paradigm has been proposed to encode every element (entity or relation) of a knowledge graph into a low-dimensional vector space [Bordes *et al.*, 2013; Socher *et al.*, 2013]. The representations of entities and relations are obtained by minimizing a global loss function involving all entities and relations. Therefore, we can do reasoning over knowledge graphs through algebraic computations.

Although these existing methods have good capability to learn knowledge graph embeddings, it remains challenging for entities with few or no facts [Ji *et al.*, 2016]. To solve the issue of KB sparsity, many methods have been proposed to learn knowledge graph embeddings by utilizing related text information [Wang *et al.*, 2014a; Zhong *et al.*, 2015; Xie *et al.*, 2016]. These methods learn joint embedding of entities, relations, and words (or phrases, sentences) into the same vector space. However, there are still three problems to be solved. (1) The optimal combination of the structural and textual representations is not well studied in these methods, in which two kinds of representations are aligned on word level or separate loss function. A good representation of an entity should jointly encode both structure and text information. (2) The text description may represent an entity from various aspects, and various relations only focus on fractional aspects of the description. A good encoder should select the information from text according to different contexts of relations since not every word in text description is useful to representing entities given a certain relation. Figure 1 gives an example of entity description in Freebase. Given a specific relation of an entity, not all information provided in its description are useful to predict the linked entities. (3) Intuitively, entities with many facts depend more on well-trained structured representation while those with few or no facts might be largely determined by text descriptions. A good representation should learn the most valuable information by weighting both sides.

In this paper, we propose a new deep architecture to learn the knowledge representation by utilizing the existing text descriptions of entities. Specifically, we learn a joint representation of each entity from two information sources: one is structure information, and another is its text description. The joint representation is the combination of the structure and text representations with a gating mechanism. The gate decides how much information from the structure or text representation will carry over to the final joint representation. In addition, we also introduce an attention mechanism to select the most related information from text description under different contexts. Experimental results on link prediction and triplet classification show that our joint models can handle the sparsity problem well and outperform the baseline method on all metrics with a large margin.

[*]Corresponding author: xpqiu@fudan.edu.cn

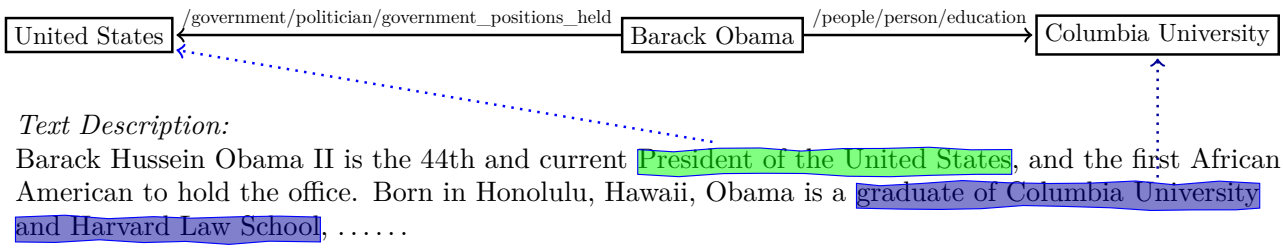

Figure 1: Example of an entity description in Freebase.

Our contributions in this paper are summarized as follows.

1. Unlike previous methods, we integrate the structure and text information of an entity into a joint representation, which can benefit the downstream applications.
2. The gate mechanism can automatically find a balance between the structure and text information. For a low-frequency entity, the description will provide supplementary information for embedding, thus the issue of sparsity in knowledge base is settled properly.
3. Given an entity, our attentive LSTM encoder can dynamically select the most related information from its text description according to different relations.

2 Knowledge Graph Embedding

In this section, we briefly introduce the background knowledge about the knowledge graph embedding.

Knowledge graph embedding aims to model multi-relational data (entities and relations) into a continuous low-dimensional vector space. Given a pair of entities (h, t) and their relation r, we can represent them with a triple (h, r, t). A score function $f(h, r, t)$ is defined to model the correctness of the triple (h, r, t), thus to distinguish whether two entities h and t are in a certain relationship r. $f(h, r, t)$ should be larger for a golden triplet (h, r, t) that corresponds to a true fact in real world, otherwise $f(h, r, t)$ should be lower for an negative triplet.

The difference among the existing methods varies between linear [Bordes *et al.*, 2013; Wang *et al.*, 2014b] and nonlinear [Socher *et al.*, 2013] score functions in the low-dimensional vector space.

Among these methods, TransE [Bordes *et al.*, 2013] is a simple and effective approach, which has achieved state-of-the-art prediction performance. It learns the vector embeddings for both entities and relationships. Its basic idea is that the relationship between two entities is supposed to correspond to a translation between the embeddings of entities, that is, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds.

TransE's score function is defined as:

$$f(h, r, t)) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2 \tag{1}$$

where $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$ are embeddings of h, t, r respectively, and satisfy $\|\mathbf{h}\|_2^2 = \|\mathbf{t}\|_2^2 = 1$. The $\mathbf{h}, \mathbf{r}, \mathbf{t}$ are indexed by a lookup table respectively.

3 Neural Text Encoding

Given an entity in most of the existing knowledge bases, there is always an available corresponding text description with valuable semantic information for this entity, which can provide beneficial supplement for entity representation.

To encode the representation of a entity from its text description, we need to encode the variable-length sentence to a fixed-length vector. There are several kinds of neural models used in sentence modeling. These models generally consist of a projection layer that maps words, sub-word units or n-grams to vector representations (often trained beforehand with unsupervised methods), and then combine them with the different architectures of neural networks, such as neural bag-of-words (NBOW), recurrent neural network (RNN) [Elman, 1990; Sutskever *et al.*, 2014; Cho *et al.*, 2014] and convolutional neural network (CNN) [Collobert *et al.*, 2011; Kalchbrenner *et al.*, 2014].

In this paper, we use three encoders (NBOW, LSTM and attentive LSTM) to model the text descriptions.

3.1 Bag-of-Words Encoder

A simple and intuitive method is the neural bag-of-words (NBOW) model, in which the representation of text can be generated by summing up its constituent word representations.

We denote the text description as a word sequence $x_{1:n} = x_1, \cdots, x_n$, where x_i is the word at position i. The NBOW encoder is

$$\text{enc}_1(x_{1:n}) = \sum_{i=1}^{n} \mathbf{x}_i, \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the word embedding of x_i.

Potentially, NBOW can capture the relative importance of words to distinguish content words from stop words or embellishments. However, the main drawback of NBOW is that the word order is lost.

3.2 LSTM Encoder

To address some of the modelling issues with NBOW, we consider using a bidirectional long short-term memory network (LSTM) [Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005] to model the text description.

LSTM was proposed by [Hochreiter and Schmidhuber, 1997] to specifically address this issue of learning long-term dependencies [Bengio *et al.*, 1994; Hochreiter *et al.*, 2001; Hochreiter and Schmidhuber, 1997] in RNN. The LSTM

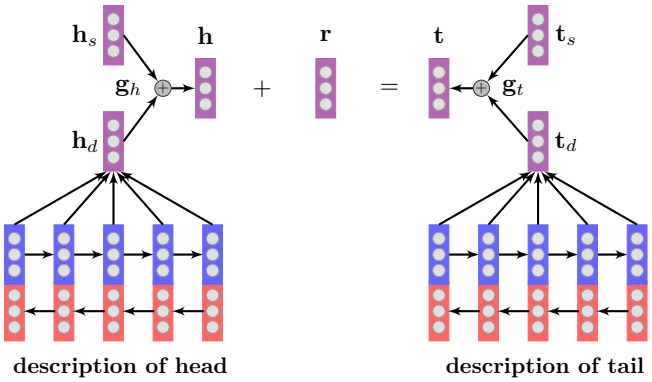

Figure 2: Our general architecture of jointly structural and textual encoding.

maintains a separate memory cell inside it that updates and exposes its content only when deemed necessary. A number of minor modifications to the standard LSTM unit have been made. While there are numerous LSTM variants, here we describe the implementation used by [Graves, 2013].

Bidirectional LSTM (BLSTM) can be regarded as two separate LSTMs with different directions. One LSTM models the text description from left to right, and another LSTM models text description from right to left respectively. We define the outputs of two LSTM at time step i are $\overrightarrow{\mathbf{z}}_i$ and $\overleftarrow{\mathbf{z}}_i$ respectively.

The combined output of BLSTM at position i is $\mathbf{z}_i = \overrightarrow{\mathbf{z}}_i \oplus \overleftarrow{\mathbf{z}}_i$, where $\oplus$ denotes the concatenation operation.

The LSTM encoder combines all the outputs $\mathbf{z}_i \in \mathbb{R}^d$ of BLSTM at different position.

$$\text{enc}_2(x_{1:n}) = \sum_{i=1}^{n} \mathbf{z}_i. \tag{3}$$

3.3 Attentive LSTM Encoder

While the LSTM encoder has richer capacity than NBOW, it produces the same representation for the entire text description regardless of its contexts. However, the text description may present an entity from various aspects, and various relations only focus on fractional aspects of the description. This phenomenon also occurs in structure embedding for an entity [Wang *et al.*, 2014b; Lin *et al.*, 2015].

Given a relation for an entity, not all of words/phrases in its text description are useful to model a specific fact. Some of them may be important for the given relation, but may be useless for other relations. Therefore, we introduce an attention mechanism [Bahdanau *et al.*, 2014] to utilize an attention-based encoder that constructs contextual text encodings according to different relations.

For each position i of the text description, the attention for a given relation r is defined as $\alpha_i(r)$, which is

$$e_i(r) = \mathbf{v}_a^T \tanh(\mathbf{W}_a \mathbf{z}_i + \mathbf{U}_a \mathbf{r}), \tag{4}$$

$$\alpha_i(r) = \mathbf{softmax}(e_i(r)) = \frac{\exp(e_i(r))}{\sum_{j=1}^{n} \exp(e_j(r))}, \tag{5}$$

where $\mathbf{r} \in \mathbb{R}^d$ is the relation embedding; $\mathbf{z}_i \in \mathbb{R}^d$ is the output of BLSTM at position i; $\mathbf{W}_a, \mathbf{U}_a \in \mathbb{R}^{d \times d}$ are parameters matrices; $\mathbf{v}_a \in \mathbb{R}^d$ is a parameter vector.

The attention $\alpha_i(r)$ is interpreted as the degree to which the network attends to partial representation $\mathbf{z}_i$ for given relation r.

The contextual encoding of text description can be formed by a weighted sum of the encoding $\mathbf{z}_i$ with attention.

$$\mathbf{enc_3}(x_{1:n}; r) = \sum_{i=1}^{n} \alpha_i(r) * \mathbf{z}_i. \tag{6}$$

4 Joint Structure and Text Encoder

Since both the structure and text description provide valuable information for an entity , we wish to integrate all these information into a joint representation.

We propose a united model to learn a joint representation of both structure and text information. The whole model can be end-to-end trained.

For an entity e, we denote $\mathbf{e}_s$ to be its embedding of structure information, $\mathbf{e}_d$ to be encoding of its text descriptions. The main concern is how to combine $\mathbf{e}_s$ and $\mathbf{e}_d$.

To integrate two kinds of representations of entities, we use gating mechanism to decide how much the joint representation depends on structure or text.

The joint representation $\mathbf{e}$ is a linear interpolation between the $\mathbf{e}_s$ and $\mathbf{e}_d$.

$$\mathbf{e} = \mathbf{g}_e \odot \mathbf{e}_s + (1 - \mathbf{g}_e) \odot \mathbf{e}_d, \tag{7}$$

where $\mathbf{g}_e$ is a gate to balance two sources information and its elements are in $[0, 1]$, and $\odot$ is an element-wise multiplication. Intuitively, when the gate is close to 0, the joint representation is forced to ignore the structure information and is the text representation only.

Gate Strategy We set $\mathbf{g}_e$ to be a static vector, which means all the dimensions of $\mathbf{e}_s$ and $\mathbf{e}_d$ are summed by the different weights. We assign a static gate $\mathbf{g}_e$ to each entity e. To constrain the value of each element is in $[0, 1]$, we use logistic sigmoid function to compute the gate.

$$\mathbf{g}_e = \sigma(\tilde{\mathbf{g}}_e), \tag{8}$$

where $\tilde{\mathbf{g}}_e \in \mathbb{R}^d$ is real-value vector and stored in a lookup table. Once $\tilde{\mathbf{g}}_e$ is learned on training data, it keeps unchanged during test.

Score Function Following TransE, our final score function is defined as

$$f(h, r, t; d_h, d_t) = \| \big(\mathbf{g}_h \odot \mathbf{h}_s + (1 - \mathbf{g}_h) \odot \mathbf{h}_d\big) + \mathbf{r} - \big(\mathbf{g}_t \odot \mathbf{t}_s + (1 - \mathbf{g}_t) \odot \mathbf{t}_d\big) \|_2^2, \tag{9}$$

where $\mathbf{g}_h$ and $\mathbf{g}_t$ are gates of head and tail respectively.

Figure 2 gives an illustration of our model. To model the structure information better, $\mathbf{h}_s$, $\mathbf{r}$, $\mathbf{t}_s$ can be pre-trained with one of existing methods of knowledge graph embeddings, such as TransE.

Dataset	#Rel	#Ent	#Train	#Valid	#Test
FB15k	1,345	14,951	483,142	50,000	59,071
WN18	18	40,493	141,442	5,000	5,000

Table 1: Statistics of datasets used in experiments.

4.1 Training

We use the contrastive max-margin criterion [Bordes *et al.*, 2013; Socher *et al.*, 2013] to train our model. Intuitively, the max-margin criterion provides an alternative to probabilistic, likelihood-based estimation methods by concentrating directly on the robustness of the decision boundary of a model [Taskar *et al.*, 2005]. The main idea is that each triplet (h, r, t) coming from the training corpus should receives a higher score than a triplet in which one of the elements is replaced with a random elements.

We assume that there are n_t triplets in training set and denote the ith triplet by $(h_i, r_i, t_i), (i = 1, 2, \cdots, n_t)$. Each triplet has a label y_i to indicate the triplet is positive ($y_i = 1$) or negative ($y_i = 0$).

Then the golden and negative triplets are denoted by $\mathcal{D} = \{(h_j, r_j, t_j)|y_j = 1\}$ and $\hat{\mathcal{D}} = \{(h_j, r_j, t_j)|y_j = 0\}$, respectively. The positive example are the triplets from training dataset, and the negative examples are generated as follows: $\hat{\mathcal{D}} = \{(h_l, r_k, t_k)|h_l \neq h_k \wedge y_k = 1\} \cup \{(h_k, r_k, t_l)|t_l \neq t_k \wedge y_k = 1\} \cup \{(h_k, r_l, t_k)|r_l \neq r_k \wedge y_k = 1\}$. The sampling strategy is Bernoulli distribution described in [Wang *et al.*, 2014b].

Let the set of all parameters be Θ, we minimize the following objective:

$$J(\Theta) = \sum_{(h,r,t)\in\mathcal{D}} \sum_{(\hat{h},\hat{r},\hat{t})\in\hat{\mathcal{D}}} \max(0, \gamma - \tag{10}$$

$$f(h, r, t) + f(\hat{h}, \hat{r}, \hat{t})\ ) + \eta\|\Theta\|_2^2, \tag{11}$$

where $\gamma > 0$ is a margin between golden triplets and negative triplets., $f(h, r, t)$ is the score function. We use the standard L_2 regularization of all the parameters, weighted by the hyperparameter η.

We use stochastic gradient descent (SGD) [Duchi *et al.*, 2011] for optimization which converges to a local optimum of our non-convex objective function.

5 Experiment

In this section, we study the empirical performance of our proposed models on two benchmark tasks: triplet classification and link prediction.

5.1 Datasets

We use two popular knowledge bases: WordNet [Miller, 1995] and Freebase [Bollacker *et al.*, 2008] in this paper. Specifically, we use WN18 (a subset of WordNet) [Bordes *et al.*, 2014] and FB15K (a subset of Freebase) [Bordes *et al.*, 2013] since their text descriptions are easily publicly available.[1]

Table 1 lists statistics of the two datasets.

[1] https://github.com/xrb92/DKRL

5.2 Link Prediction

Link prediction is a subtask of knowledge graph completion to complete a triplet (h, r, t) with h or t missing, i.e., predict t given (h, r) or predict h given (r, t). Rather than requiring one best answer, this task emphasizes more on ranking a set of candidate entities from the knowledge graph.

Similar to [Bordes *et al.*, 2013], we use two measures as our evaluation metrics. (1) Mean Rank: the averaged rank of correct entities or relations; (2) Hits@p: the proportion of valid entities or relations ranked in top p predictions. Here, we set $p = 10$ for entities and $p = 1$ for relations. A lower Mean Rank and a higher Hits@p should be achieved by a good embedding model. We call this evaluation setting "Raw". Since a false predicted triplet may also exist in knowledge graphs, it should be regard as a valid triplet. Hence, we should remove the false predicted triplets included in training, validation and test sets before ranking (except the test triplet of interest). We call this evaluation setting "Filter". The evaluation results are reported under these two settings.

Implementation We select the margin γ among $\{1, 2, 10\}$, the embedding dimension d among $\{20, 50, 100\}$, the regularization η among $\{0, 1E{-}5, 1E{-}6\}$, two learning rates λ_s and λ_t among $\{0.001, 0.01, 0.05, 0.1\}$ to learn the parameters of structure and text encoding. The dissimilarity measure is set either to the L_1 or L_2 distance. The best configurations obtained by validation set are chosen for the evaluation. For all data sets, training time was limited to at most $1,000$ epochs over the training set.

In order to speed up the convergence and avoid overfitting, we initiate the structure embeddings of entity and relation with the results of TransE. The embedding of a word is initialized by averaging the linked entity embeddings whose description include this word. The rest parameters are initialized by randomly sampling from uniform distribution in $[-0.1, 0.1]$.

The final optimal configurations are: $\gamma = 2$, $d = 20$, $\eta = 1E{-}5$, $\lambda_s = 0.01$, $\lambda_t = 0.1$, and L_1 distance on WN18; $\gamma = 2$, $d = 100$, $\eta = 1E{-}5$, $\lambda_s = 0.01$, $\lambda_t = 0.05$, and L_1 distance on FB15K.

Results Experimental results on both WN18 and FB15k are shown in Table 2, where we use "Jointly(CBOW)", "Jointly(LSTM)" and "Jointly(A-LSTM)" to represent our jointly encoding models with CBOW, LSTM and attentive LSTM text encoders. Our baseline is TransE since that the score function of our models is based on TransE.

From the results, we observe that our proposed jointly structural and textual encoding models are much better than the baseline TransE on all metrics, which indicates that

Datasets	WN18				FB15K			
Metric	Mean Rank		Hits@10		Mean Rank		Hits@10	
	Raw	Filt	Raw	Filt	Raw	Filt	Raw	Filt
Unstructured [Bordes *et al.*, 2012]	315	304	35.3	38.2	1,074	979	4.5	6.3
SME (linear) [Bordes *et al.*, 2012]	545	533	65.1	74.1	274	154	30.7	40.8
SME (Bilinear) [Bordes *et al.*, 2012]	526	509	54.7	61.3	284	158	31.3	41.3
TransH [Wang *et al.*, 2014b]	318	303	75.4	86.7	212	87	45.7	64.4
TransR [Lin *et al.*, 2015]	238	225	**79.8**	**92.0**	198	77	48.2	68.7
TransD [Ji *et al.*, 2015]	224	212	**79.6**	**92.2**	194	91	**53.4**	**77.3**
CNN+TransE [Xie *et al.*, 2016]	-	-	-	-	181	91	49.6	67.4
TransE (Baseline)	263	251	75.4	89.2	243	125	34.9	47.1
Jointly(CBOW)	142	130	78.5	89.9	183	92	48.9	67.4
Jointly(LSTM)	**117**	**95**	**79.5**	**91.6**	179	90	49.3	69.7
Jointly(A-LSTM)	134	123	78.6	90.9	**167**	**77**	**52.9**	**75.5**

Table 2: Results on link prediction.

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 (Baseline)	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
Jointly(CBOW)	75.4	91.6	18.5	44.1	75.2	24.6	92.2	52.3
Jointly(LSTM)	81.3	88.9	18.8	45.2	80.1	25.4	89.6	52.4
Jointly(A-LSTM)	**83.8**	**95.1**	**21.1**	**47.9**	**83**	**30.8**	**94.7**	**53.1**

Table 3: Detailed results by category of relationship on FB15K.

knowledge representation can benefit greatly from text description.

On WN18, "Jointly(LSTM)" achieves the best performances, and the "jointly(A-LSTM)" is slightly worse than "Jointly(LSTM)". The reason is that the number of relations is relatively small. Therefore, the attention mechanism does not have obvious advantage. On FB15K, "Jointly(A-LSTM)" achieves the best performance and is significantly higher than state-of-the-art methods on mean rank.

Although the Hits@10 of our models are worse than the best state-of-the-art method, TransD[Ji *et al.*, 2015], it is worth noticing that the score function of our models is based on TransE, not TransD. Our models are compatible with other state-of-the-art knowledge embedding models. We believe that our model can be further improved by adopting the score functions of other state-of-the-art methods, such as TransD.

Besides, textual information largely alleviates the issue of sparsity and our model achieves substantial improvement on Mean Rank comparing with TransD. However, textual information may slightly degrade the representation of frequent entities which have been well-trained. This may be another reason why our Hits@10 is worse than TransD which only utilizes structural information.

Detailed Results by Frequency of Entity To further analyze these results, we analyse the performances of our joint encoding models according to the different frequencies of entities. We sort the entities by their frequencies and partition them into ten equal-size groups.

Table 3 shows Hit@10(Filter) of the different models on these groups. We can find that our model consistently outperforms the baseline TransE on all the groups, which indicates that the text description benefits not only low frequency entity, but also high frequency entity.

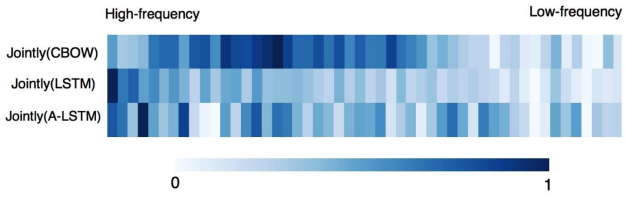

Figure 3: Visualization of gates with different entity frequencies on FB15K.

Visualization of Gates To get more insights into how the joint representation is influenced by the structure and text information. We observe the activations of gates, which control the balance between two sources of information, to understand the behavior of neurons. We sort the entities by their frequencies and divide them into 50 equal-size groups of different frequencies, and average the values of all gates in each group.

Figure 3 gives the average of gates in ten groups from high- to low-frequency. We observe that the text information play more important role for the low-frequency entities.

5.3 Triplet Classification

Triplet classification is a binary classification task, which aims to judge whether a given triplet (h, r, t) is a correct fact or not. Since our used test sets (WN18 and FB15K) only contain correct triplets, we construct negative triplets following the same setting used in [Socher *et al.*, 2013].

For triplets classification, we set a threshold δ_r for each relation r. δ_r is obtained by maximizing the classification accuracies on the valid set. For a given triplet (h, r, t), if its score is larger than δ_r, it will be classified as positive, otherwise negative.

Datasets	WN18	FB15K
TransE	92.9	79.8
TransH	-	79.9
TransR	-	82.1
CTransR	-	84.3
TransD	-	88.0
TranSparse	-	88.5
Jointly(NBOW)	97.5	89.7
Jointly(LSTM)	97.7	90.5
Jointly(A-LSTM)	**97.8**	**91.5**

Table 4: Results on triplet classification.

Results Table 4 shows the evaluation results of triplets classification. The results reveal that our joint encoding models is effective and also outperform the state-of-the-art methods.

On WN18, "Jointly(A-LSTM)" achieves the best performance, and the "Jointly(LSTM)" is slightly worse than "Jointly(A-LSTM)". The reason is that the number of relations is relatively small. Therefore, the attention mechanism does not show obvious advantage. On FB15K, the classification accuracy of "Jointly(A-LSTM)" achieves 91.5%, which is the best and significantly higher than that of state-of-the-art methods.

6 Related Work

Recently, it has gained lots of interests to jointly learn the embeddings of knowledge graph and text information. There are several methods using textual information to help KG representation learning.

[Socher *et al.*, 2013] represent an entity as the average of its word embeddings in entity name, allowing the sharing of textual information located in similar entity names.

[Wang *et al.*, 2014a] jointly embed knowledge and text into the same space by aligning the entity name and its Wikipedia anchor, which brings promising improvements to the accuracy of predicting facts. [Zhong *et al.*, 2015] extend the joint model and aligns knowledge and words in the entity descriptions. However, these two works align the two kinds of embeddings on word level, which can lose some semantic information on phrase or sentence level.

[Zhang *et al.*, 2015] also represent entities with entity names or the average of word embeddings in descriptions. However, their use of descriptions neglects word orders, and the use of entity names struggles with ambiguity. [Xie *et al.*, 2016] jointly learn knowledge graph embeddings with entity descriptions. They use continuous bag-of-words and convolutional neural network to encode semantics of entity descriptions. However, they separate the objective functions into two energy functions of structure-based and description-based representations. To utilize both representations, they need further estimate an optimum weight coefficients to combine them together in the specific tasks.

Besides entity representation, there are also a lot of works [Lao *et al.*, 2012; Toutanova *et al.*, 2015; Neelakantan *et al.*, 2015] to map textual relations and knowledge base relations to the same vector space and obtained substantial improvements.

7 Conclusion

We propose a united representation for knowledge graph, utilizing both structure and text description information of the entities. Experiments show that our proposed jointly representation learning with gating mechanism is effective, which benefits to modeling the meaning of an entity.

In the future, we will consider the following research directions to improve our model:

1. Currently, our score function is based on TransE since the main emphasis of this work is how to integrate both structural and textual information. We believe our models can be further improved with the recently proposed knowledge graph embedding models, such as TransH [Wang *et al.*, 2014b], TransR [Lin *et al.*, 2015], TransD [Ji *et al.*, 2015] and so on.
2. We will try to use dynamical gating strategy to integrate the representations of structure and text into a unified architecture. The gate can be estimated according to the context information, which can reduce the parameters greatly.
3. Intuitively, text descriptions of relations are also helpful to learn better semantic representations. Therefore, we will extend our model to incorporate text descriptions of both entities and relations.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was partially funded by Shanghai Municipal Science and Technology Commission (No. 16JC1420401), National Natural Science Foundation of China (No. 61532011 and 61672162).

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Bengio *et al.*, 1994] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[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 the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[Bordes *et al.*, 2012] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *AISTATS*, 2012.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.

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

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*, 2014.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, 2011.

[Duchi *et al.*, 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2011.

[Elman, 1990] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[Graves and Schmidhuber, 2005] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[Graves, 2013] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hochreiter *et al.*, 2001] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[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*, 2015.

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

[Kalchbrenner *et al.*, 2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *ACL*, 2014.

[Lao *et al.*, 2012] Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W Cohen. Reading the web with learned syntactic-semantic inference rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1017–1026. Association for Computational Linguistics, 2012.

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

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

[Neelakantan *et al.*, 2015] Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. Compositional vector space models for knowledge base completion. *arXiv preprint arXiv:1504.06662*, 2015.

[Schuster and Paliwal, 1997] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 1997.

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

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in NIPS*, pages 3104–3112, 2014.

[Taskar *et al.*, 2005] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of the ICML*, 2005.

[Toutanova *et al.*, 2015] Kristina Toutanova, Danqi Chen, Patrick Pantel, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of EMNLP*, 2015.

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

[Wang *et al.*, 2014b] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, 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 *Proceedings of IJCAI*, 2016.

[Zhang *et al.*, 2015] Dongxu Zhang, Bin Yuan, Dong Wang, and Rong Liu. Joint semantic relevance learning with text data and graph knowledge. *ACL-IJCNLP 2015*, 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 *Proceedings of EMNLP*, 2015.