REPRESENTATION LEARNING OF KNOWLEDGE GRAPHS USING CONVOLUTIONAL NEURAL NETWORKS

W. Gao[*], *Y. Fang*[†], *F. Zhang*[‡], *Z. Yang*[§]

Abstract: Knowledge graphs have been playing an important role in many Artificial Intelligence (AI) applications such as entity linking, question answering and so forth. However, most of previous studies focused on the symbolic representation of knowledge graphs with structural information, which cannot deal well with new entities or rare entities with little relevant knowledge. In this paper, we propose a new deep knowledge representation architecture that jointly encodes both structure and textual information. We first propose a novel neural model to encode the text descriptions of entities based on Convolutional Neural Networks (CNN). Secondly, an attention mechanism is applied to capture the valuable information from these descriptions. Then we introduce position vectors as supplementary information. Finally, a gate mechanism is designed to integrate representations of structure and text into the joint representation. Experimental results on two datasets show that our models obtain state-of-the-art results on link prediction and triplet classification tasks, and achieve the best performance on the relation classification task.

Key words: *convolutional neural network, knowledge representation learning, knowledge base, attention*

Received: December 19, 2018
Revised and accepted: June 30, 2020

DOI: 10.14311/NNW.2020.30.011

1. Introduction

At present, knowledge bases have broad application prospects in the field of artificial intelligence such as question answering and sentiment analysis [5, 26]. A knowledge base is usually represented as a network structure, using triplets *(Head Entity, Relation, Tail Entity)* to represent knowledge. For instance, we know that

[*] Wang Gao – Corresponding author; School of Artificial Intelligence, Jianghan University, Wuhan, China; School of Computer Science, Wuhan University, Wuhan, China, E-mail: gaowang2000@foxmail.com

[†] Yuan Fang; School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China, E-mail: fangyuan2000@foxmail.com

[‡] Fan Zhang; College of Computer Science, Wuhan Donghu University, Wuhan, China, E-mail: whzhangfan@126.com

[§] Zhifeng Yang; College of Sports Science and Technology, Wuhan Sports University, Wuhan, China, E-mail: yangzhif@foxmail.com

Moscow is the capital of Russia. In knowledge bases, we will represent this fact with the triple form as (Moscow, $is\_capital\_of$, Russia). However, network-based knowledge representation methods face the following challenges: (1) Low computational efficiency. To make use of the knowledge in network-based knowledge bases, it is often necessary to design specialized graph algorithms. However, these algorithms are not suitable for knowledge reasoning. As the size of the knowledge base increases, they usually encounter problems of low computational inefficiency and the lack of scalability [15]. (2) Severe data sparsity. There are some rare entities in the knowledge base with less associated knowledge, and their semantic calculation accuracy is extremely low [24]. To solve the above challenges, researchers have proposed knowledge representation learning methods based on deep learning, in which TransE is the most widely used model [3]. However, TransE and most of its extended models only use the structural information of knowledge bases, which makes them difficult to deal with new entities or rare entities with little relevant knowledge. The reason is that these entities have no or little structured information available [21].

To tackle the data sparsity problem, many studies have begun to introduce textual information to improve knowledge representation [21, 28, 23]. For new or rare entities, they employ text descriptions to supplement semantic information, which can effectively alleviate the data sparsity problem. However, this strategy still has following shortcomings: (1) Effective ways of combining textual and structural information have not been proposed. Many studies only align on word level or the score function. (2) Textual information is not filtered. Not every word in a text description helps to represent an entity with a specific relation.

Recently, Xu et al. proposed a joint knowledge representation model based on bi-directional Long Short-Term Memory (LSTM) [24]. The model employs attention to filter the information in text descriptions, and proposes a gate mechanism to combine textual and structure representations. It achieves the best performance in classical tasks such as link prediction and triplet classification. However, the bi-directional LSTM model needs to input the previous hidden state and position to generate the next hidden state. This inherently sequential nature prevents the training process from being parallelized. When processing continuous sequences, the cross-batch processing of the training set is restricted due to memory constraints [20].

To address the above problems and challenges, this paper proposes an attention-based CNN joint knowledge representation model, referred to as ACNNM. The model accurately captures the most relevant semantics in the text description, mitigating the sparse problem of knowledge bases. Furthermore, ACNNM exploits the advantages of convolution kernels for parallelization and efficient computation. Specifically, we first propose a text encoder based on CNN, and then design a special attention mechanism to select the semantic information that is most relevant to the entity in the description. Secondly, ACNNM uses the TransE model to encode the structure information of knowledge bases. Finally, this paper introduces a gate mechanism to find a balance between structure and textual information. In addition, we also propose an extended model PACNNM based on ACNNM. The PACNNM model attempts to introduce a position vector at the input, so that the encoder is able to capture the position information of words. Experiments on link prediction and triplet classification tasks show that the proposed model can significantly improve the sparse problem. The results are highly competitive compared with the state-of-the-art baselines, especially on the relation classification task. The main contributions of this paper are as follows:

1. This paper proposes an attention-based CNN joint knowledge representation model ACNNM for combining entity descriptions and structure information. ACNNM designs a special attention mechanism to capture the most relevant information in descriptions, which helps to improve the discrimination of entity representation. To the best of our knowledge, this work is the first to encode both structural and textual information of entities by using the attention-based CNN model.

2. This paper also proposes an extended model PACNNM based on ACNNM. The model introduces a position vector that enables CNN to capture positional information in text descriptions.

3. Experimental results demonstrate that the proposed models achieve comparable performance against the state-of-the-art baselines on link prediction and triplet classification tasks, and achieve the best performance on the relation classification task.

2. Related work

Knowledge representation learning aims to learn the distributed representations of entities and relations in knowledge graph, and project their representations into a low-dimensional continuous semantic space, which has been widely utilized in many knowledge-driven tasks. Unlike traditional representation methods, knowledge representation learning provides much dense representations of entities and relations, thereby reducing the computational complexity of its applications.

Attention mechanisms in neural networks (also called neural attention or attention only) have been successfully applied to various natural language processing tasks. The attention mechanism provides a neural network with the ability to focus on a subset of its inputs (or features), that is, it can select specific inputs to improve the performance of the neural network.

In this section, we briefly summarize the related work from the following two perspectives: knowledge representation learning and knowledge representation by introducing textual information.

2.1 Knowledge representation learning

In recent years, knowledge representation learning has been a research hotspot due to its excellent performance in various tasks such as knowledge acquisition, integration, reasoning and topic evolution analysis [16, 10, 7, 9].

Bordes et al. proposed an unstructured model, which embed head and tail entities $(\mathbf{h}, \mathbf{t})$ into a vector space. The model assumes that head and tail entity vectors are similar (*i.e.*, $||\mathbf{h} - \mathbf{t}||_2^2 \approx 0$), and relation vectors are set to zero in the score function [1]. Therefore, it is impossible to distinguish different relations. Based on the unstructured model, Bordes et al. proposed a structured embedding model, which assumes that head and tail entity vectors are similar only in the semantic space of correlated relations [4]. Furthermore, Bordes et al. proposed a semantic matching energy model, which uses a projection matrix to represent entities and relations, and divides them into linear and bilinear forms according to a score function [2]. After that, Bordes et al. proposed the TransE model, which is very efficient while achieving the state-of-the-art predictive performance [3]. TransE has therefore gradually become the most concerned knowledge representation model.

TransE represents the relation as a translation vector from the head entity to the tail entity, aiming to project entities and relations into the same low-dimensional vector space. However, entities cannot have different representations under different relations in the TransE model. To solve the problem, Wang et al. proposed the TransH model, which models the relation as a hyperplane and projected the head and tail entity to a hyperplane with a specific relation [21]. Unlike TransE and TransH assuming that entities and relations are in the same vector space, Lin et al. proposed the TransR model, which represents entities and relations in different semantic spaces, and embeds entities into corresponding relational spaces [14]. Based on TransR, Lin et al. further proposed the CTransR model, which uses the clustering relationship to learn the representation vector for each relation [14]. To solve the problem of too many parameters in TransR, Ji et al. proposed the TransD model [11]. By considering the diversity of both entities and relations, TransD creates a dynamic mapping matrix for each entity-relation pair. In addition, Ji et al. proposed the TranSparse model, replacing the dense matrix in the TransR model with a sparse matrix [12]. Both head and tail entities have projection matrices, where the number of connected entities determines the sparsity of the matrix.

However, the above methods only utilize the structural information of knowledge bases, and fail to employ other information related to the knowledge base, such as entity descriptions. By the contrast, the proposed models combine entity descriptions with structural information to mitigate the sparse problem of knowledge bases.

2.2 Knowledge representation by introducing textual information

At present, many researchers try to integrate textual information to alleviate data sparseness and improve knowledge representation. For instance, Socher et al. proposed a knowledge representation model that represents an entity as the average of word embeddings in the entity name [18]. By aligning entity names with Wikipedia anchors, Wang et al. proposed a novel method that embeds knowledge and texts into the same space to improve the accuracy of predicting facts [22]. Based on this method, Zhong et al. extended the model to relate the knowledge and vocabulary in entity descriptions [28]. However, the above methods align the two kinds of embeddings on word level, resulting in the loss of semantic information on phrase or sentence level. Zhang et al. use entity names or the average of word embeddings in descriptions to represent entities, which ignores word order information in the sentence [27].

Xie et al. jointly learn knowledge graph embeddings with entity descriptions [23]. They utilize CNN and continuous bag-of-words to encode semantics of entity descriptions, and divides the score function into two parts based on structure and description information, respectively. Although their method also exploits CNN to encode textual information, its CNN only includes a convolution layer, a non-linear layer and a pooling layer, which is very different from the CNN structure of the proposed models. Xu et al. proposed a joint representation model based on bi-directional LSTM [24]. An attention mechanism is used to select the relevant text in entity descriptions, and a gate mechanism is designed to control the weight of structural information and textual information. However, the hidden state of the bi-directional LSTM model needs to be generated sequentially, and cannot be processed in parallel during the training process, which limits the computational efficiency of long sequences. In contrast, our models employ an attention-based CNN to model both structure and textual information. Since the weights of neurons on the same feature map are the same, CNN is very suitable for parallel processing. To the best of our knowledge, this work is the first to encode both structural and textual information of entities by using the attention-based CNN model.

3. Methodology

The joint model of this paper is mainly divided into three parts: structure representation based on TransE, text representation based on CNN/ACNN/PACNN and multi-source information fusion based on the gate mechanism. We first utilize TransE to encode the structure information of triplets. Secondly, three text encoders encoding the entity description are designed: CNN, CNN with an attention mechanism (ACCN) and ACCN with position information (PACNN). Finally, a gate mechanism is used to determine the weight of the joint representation of structure representation and text representation. Fig. 1 shows the overall framework of the joint knowledge representation model. The functions of each layer of the model are described in detail below.

3.1 Structure representation based on TransE

TransE-based representation models perform well in tasks such as knowledge reasoning, relationship extraction and so forth [6, 25]. Given a triplet *(Head Entity, Relation, Tail Entity)*, it is represented as (h, r, t). The vector corresponding to triplet (h, r, t) is represented as $(\mathbf{h}, \mathbf{r}, \mathbf{t})$. TransE is designed to represent entities and relations as low-dimensional continuous vectors. The vector of a positive triplet should satisfy the formula $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, and negative triplets are not satisfied. A score function $f_r(h, r, t)$ is used to distinguish whether two entities h and t are in a certain relationship r, and to model the correctness of the triplet (h, r, t). For a positive triplet (h, r, t) corresponding to a true fact in real world, $f_r(h, r, t)$ should be larger, otherwise for an negative triplet, $f_r(h, r, t)$ should be lower. Therefore, TransE defines the following score function to measure the quality of triplets:

$$f_r(h, r, t) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||_{L_1/L_2}, \qquad s.t., ||\mathbf{h}||_2^2 \leqslant 1; ||\mathbf{t}||_2^2 \leqslant 1. \tag{1}$$

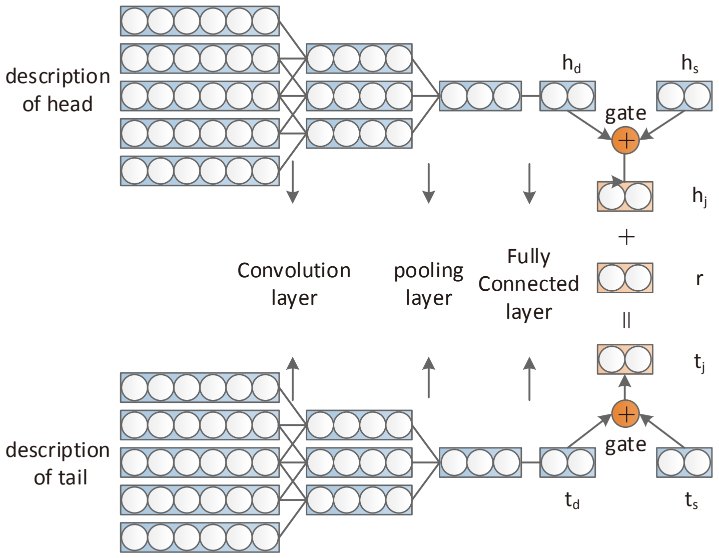

Fig. 1 *The overall framework of our joint knowledge representation model.*

Eq. 1 is the L_1 or L_2 distance of vectors $\mathbf{h}+\mathbf{r}$ and $\mathbf{t}$. For a reasonable score function, the scores of positive triplets are lower than negative triplets. For a triplet (h, r, t), we denote $\mathbf{h}_s$ and $\mathbf{t}_s$ to be its head and tail embeddings of structure information learned by the TransE model, respectively.

3.2 Text representation based on CNN/ACNN/PACNN

Currently, entities in large knowledge bases usually have their corresponding entity description information. Entity descriptions contain semantic information of entities in various scenarios, which help to improve entity representations and make them more discriminative. In this paper, we need to encode textual information from entity descriptions of different lengths. Convolution kernels can capture the local features of textual information, and CNN can be parallelized and efficiently calculated. This paper therefore chooses the CNN-based text encoding method.

3.2.1 Text representation based on CNN

Preprocessing. The proposed model first removes the punctuation in entity descriptions, and then each word is represented as a pre-trained word2vec embedding [17].

Convolution layer. Let $x_{1:n} = x_1, x_2, \ldots, x_n$ be the words of a sentence of length n. $x_i \in \mathbb{R}^d$ denotes the word vector corresponding to the i^{th} word in the d-dimensional space. For a sentence x, the convolution layer selects a sequence of words using a sliding window with size h to perform a convolution operation, and outputs a feature map c. Due to the length of sentences is not fixed, this paper takes the maximum length N as the standard. Zero paddings are added at the end of a sentence if it has less than N words. The sequence of words processed by the sliding window is defined as:

$$x_{i:i+h-1} = x_i, x_{i+1}, \ldots, x_{i+h-1}. \tag{2}$$

A convolution operation involves a filter $\mathbf{w} \in \mathbb{R}^{h \times d}$, which is applied to a window of h words to generate a new feature. For instance, a feature c_i is generated from a window of words $x_{i:i+h-1}$ by:

$$c_i = f(\mathbf{w} \cdot x_{i:i+h-1} + b), \tag{3}$$

where $b \in \mathbb{R}$ is a bias term and f is an activation function. In this paper, we choose ReLU activation function, which has the merits of a non-saturating form compared to Sigmoid.

We use the same zero-padding strategy as in [13]. The output of the convolutional layer is:

$$\mathbf{c} = [c_1, \ldots, c_{n-h+1}]. \tag{4}$$

Pooling layer. Pooling (max, min, average *etc.*) is commonly used to extract robust features from convolution. In this paper, average pooling is applied to each feature map to induce a fixed-length vector. The i^{th} vector of the pooling layer output with window size h is:

$$\hat{c}_i = \text{ average}(c_{i \cdot 1}, \ldots, c_{i \cdot n-h+1}). \tag{5}$$

The output of the pooling layer is $\hat{\mathbf{c}} = [\hat{c}_1, \ldots, \hat{c}_m]$, where m denotes the number of convolutional filters.

Dropout. Dropout is a technique to prevent neural networks from overfitting and approximate a way to combine exponentially different neural network architectures [19]. When training the model, the hidden unit has a probability to be temporarily removed from the network, which can be sampled by a Bernoulli distribution. The output of the dropout layer is defined as:

$$\hat{\mathbf{c}}^* = \mathbf{r} \odot \hat{\mathbf{c}}, \mathbf{r} \sim \text{ Bernoulli}(\rho), \tag{6}$$

where $\odot$ is the element-wise multiplication operator, and $\mathbf{r}$ is a vector of 0 or 1 generated by a Bernoulli distribution with parameter ρ. These dropout units will be ignored when computing input and output both in the progress of forward and backward propagation.

Fully connected layers. Neurons in this layer have full connections with all neurons in the previous layer to obtain the final output vector of the network. The output of CNN is defined as:

$$\mathbf{e_d} = \mathbf{w_o} \cdot \hat{\mathbf{c}}^* + b_o, \tag{7}$$

where $\mathbf{w_o}$ is the convolution weights, b_o is the bias. We denote $\mathbf{h_d}$ and $\mathbf{t_d}$ to be head and tail embeddings of textual information.

3.2.2 Text representation based on ACNN

CNN encodes the whole text description, without considering that the description information contains different semantics of the entity under various relations. This means that given a triplet including a particular relation, information about other relations contained in the description will be disturbed. Therefore, we proposes a new text encoder ACNN based on CNN. ACNN designs an attention mechanism that utilizes the relation of triplets to capture the most relevant information in the description.

For a word sequence of the entity description $x_{1:n} = x_1, x_2, \ldots, x_n$, when relation $\mathbf{r} \in R^d$ is given, the attention of the description is defined as:

$$a(\mathbf{r}) = \text{ Softmax}(x_{1:n} \cdot \mathbf{r}). \tag{8}$$

Suppose the output of the convolutional layer is $\mathbf{c}$, and after adding the attention weight, the output becomes $\mathbf{c}^*$, which will be used as the input to the pooling layer. $\mathbf{c}^*$ is defined as follow:

$$\mathbf{c}^* = \mathbf{c} \cdot a(\mathbf{r}). \tag{9}$$

3.2.3 Text representation based on PACNN

Since ACNN encodes a text without considering the sequential feature of words, word order information is lost. This paper proposes PACNN based on ACNN, which introduces the positional encoding of words as supplementary information. The j^{th} component I_j of the input vector I consists of the component p_j of the position vector and the component x_j of the word vector. The position vector p_j is a column vector with the following structure:

$$p_{ij} = (1 - j/n) - (i/d)(1 - 2j/n), \tag{10}$$

where n is the number of words in the sentence, d is the dimension of the position vector, i denotes the i^{th} component of p_j. Position vectors use the same dimension as word vectors, which makes it easy to add them together.

Given a sequence of words $x_{1:n} = x_1, x_2, \ldots, x_n$, the position vector is $p_{1:n} = p_1, p_2, \ldots, p_n$. After adding the position information, the new input of the encoder is $I_{1:n} = x_1 + p_1, x_2 + p_2, \ldots, x_n + p_n$.

3.3 Multi-source information fusion

Both structural information and textual description provide effective information for entities. In this paper, we employ a gate mechanism to integrate two information sources into a joint representation $\mathbf{e}_j$, which is treated as the weighted summation of structure representation $\mathbf{e}_s$ and text representation $\mathbf{e}_d$. Joint representation $\mathbf{e}_j$ is defined as:

$$\mathbf{e}_j = \mathbf{g}_s \odot \mathbf{e}_s + \mathbf{g}_d \odot \mathbf{e}_d, \qquad s.t., \mathbf{g}_d = 1 - \mathbf{g}_s; \mathbf{g}_d, \mathbf{g}_s \in [0, 1], \tag{11}$$

where $\mathbf{g}_s$ and $\mathbf{g}_d$ are the gates that balance two sources of information, $\odot$ denotes an element-wise multiplication. We use logistic sigmoid function σ to calculate the gate $\mathbf{g}$.

$$\mathbf{g} = \sigma(\widetilde{\mathbf{g}}), \tag{12}$$

where $\widetilde{\mathbf{g}} \sim Uniform(0,1)$ is a real-valued vector and stored in a lookup table. σ is employed to constrain the value of each element between $[0,1]$. Once $\widetilde{\mathbf{g}}$ is learned on training dataset, it will remain unchanged during the test. Similar to TransE, the joint representation of the score function is defined as:

$$f_r(h,r,t) = ||(\mathbf{g}_{hs} \odot \mathbf{h}_s + \mathbf{g}_{hd} \odot \mathbf{h}_d) + \mathbf{r} - (\mathbf{g}_{ts} \odot \mathbf{t}_s + \mathbf{g}_{td} \odot \mathbf{t}_d)||^2_{L_1/L_2}, \tag{13}$$

where $\mathbf{g}_{hs}$ and $\mathbf{g}_{hd}$ are the gates of head entities, and $\mathbf{g}_{ts}$ and $\mathbf{g}_{td}$ represent the gates of tail entities.

3.4 Training

Following [3], we also exploit the maximum interval method [3, 18] to train our model. In this paper, score function $f_r(h,r,t)$ is used to evaluate the quality of triplets. The main idea is that each triplet from the training corpus should receive a higher score than a triplet with one element replaced by a random element. Therefore, we minimize the following objective function:

$$L = \sum_{(h,r,s)\in\mathcal{X}} \sum_{(h',r',s')\in\mathcal{X}'} \max(0, f_r(h,r,t) + \gamma - f_r(h^{'},r^{'},t^{'})), \tag{14}$$

where $\mathcal{X}$ is a set of positive triplets, $\mathcal{X}^{'}$ denotes a set of negative triplets and $\gamma > 0$ is a margin hyperparameter between positive triplets and negative triplets. We use stochastic gradient descent (SGD) to optimize the objective function.

The triplets in knowledge bases are all positive samples, and thus negative triplets need to be generated. Following the sampling strategy described in [22], we set different probabilities for replacing the head or tail entity, when corrupting the triplet. This method divides the relationship into four types: 1-to-1, 1-to-N, N-to-1 and N-to-N according to the number of connected entities at both ends. If the relation is 1-to-N, we tend to give more chance to replace the head entity, and give more chance to replace the tail entity if the relation is N-to-1. The rationale behind this approach is that the chance of generating false negative labels is reduced.

For each triplet, positive samples are represented by $\mathcal{X} = \{(h_i, r_i, t_i)|y_i = 1\}$ and negative samples are represented by $\mathcal{X}^{'} = \{(h^{'}_i, r^{'}_i, t^{'}_i)|y_i = -1\}$. The negative samples in the training set are generated as follows:

$$\mathcal{X}^{'} = \{(h_{neg}, r_k, t_k)|h_{neg} \neq h_k \wedge y_k = -1\} \cup \{(h_k, r_k, t_{neg}|t_{neg} \neq t_k \cup y_k = -1)\}. \tag{15}$$

4. Experiment

In this section, we conduct experiments to demonstrate the effectiveness of our proposed models against the state-of-the-art baseline methods on two benchmark tasks: link prediction and triplet classification.

4.1 Datasets

In our experiments, we use two popular knowledge bases: WordNet and Freebase, which have been used in a few studies [3, 4]. Specifically, we use WN18 (a subset of WordNet) and FB15K (a relatively dense subgraph of Freebase) because their text descriptions are easily publicly available[1]. Please see Tab. I for more details of the two datasets. #Rel and #Ent represent the number of relations and entities. #Train, #Vaild and #Test represent the size of training, validation and test datasets, respectively.

Dataset	#Rel	#Ent	#Train	#Valid	#Test
WN18	18	40,493	141,442	5,000	5,000
FB15K	1,345	14,951	483,142	50,000	59,071

Tab. I *Statistics of datasets used in the experiments.*

4.2 Baseline methods

Baseline methods are divided into three categories: (1) The models proposed in this paper: CNNM, ACNNM and PACNNM. (2) Knowledge representation models using only structural information: TransE [3], TransH [22], TransR [14], CTransR [14], TransD [11], TranSparse [12], SME(linear) [2], SME(Bilinear) [2] and Unstructured [1]. (3) Knowledge representation models using textual information: Jointly(LSTM) [24], Jointly(A-LSTM) [24] and CNN+TransE [8]. Since the datasets are the same, we directly copy experimental results of several baselines from [24].

We set the embedding dimension d in $\{50, 100, 150\}$, the margin γ in $\{0.1, 1, 3, 5, 10\}$, the learning rate λ in $\{0.000001, 0.0001, 0.01, 0.1, 1\}$, the number of filters f in $\{16, 32, 64, 128\}$, the size of kernel s in $\{1, 2, 3, 4, 5\}$. The dropout rate is set to 0.5 and the similarity measure L is set either to the L_1 or L_2 distance. To accelerate convergence and avoid overfitting, the results of TransE are used to initialize the structure embeddings of entities and relations.

In the experiment, CNNM, ACNNM and PACNNM share the same set of optimal hyperparameters. In the link prediction task, the optimal hyperparameters of the proposed models are: $d = 150, \gamma = 2, \lambda = 0.0001, f = 64, s = 5, L = L_1$ on FB15K; $d = 50, \gamma = 5, \lambda = 0.0001, f = 64, s = 5, L = L_1$ on WN18. In the triplet classification task, the optimal hyperparameters are: $d = 150, \gamma = 1, \lambda = 0.1, f = 16, s = 1, L = L_1$ on FB15K; $d = 50, \gamma = 0.1, \lambda = 0.1, f = 64, s = 5, L = L_1$ on WN18.

4.3 Link prediction

The link prediction task is designed to predict missing head or tail entities in triplets. For each valid triple, we first replace its head or tail entity with other entities. Then we calculate the scores of corrupted triplets, sort the scores in ascending order, and finally record the rank of triplets. Following [3], two measures as our evaluation metrics are reported. (1) Mean: the averaged rank of valid entities; (2) Hits@10: the proportion of correct entities ranked in top 10 predictions.

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

A good representation model should have a lower Mean and a higher Hits@10 value under this task.

Following [22], the above evaluation setting is called "Raw". Triplets may also be valid after replacing head or tail entities. It is unreasonable that such corrupted triplets may be ranked in front of positive triplets. Therefore, such false predicted triplets in training, test and validation sets should be removed before ranking. This evaluation setting is called "Filt". Experimental results on both datasets under these two settings are shown in Tab. II.

Datasets	WN18				FB15K			
Metric	Mean Raw		Hits@10 Filt		Mean Raw		Hits@10 Filt	
TransE (Baseline)	263	251	75.4	89.2	243	125	34.9	47.1
TransH	401	388	73.0	82.3	212	87	45.7	64.4
TransR	238	225	**79.8**	92.0	198	77	48.2	68.7
CTransR	231	218	79.4	92.3	199	75	48.4	70.2
TransD	224	212	79.6	**92.2**	194	91	**53.4**	**77.3**
SME(linear)	545	533	65.1	74.1	274	154	30.7	40.8
SME(Bilinear)	526	509	54.7	61.3	284	158	31.3	41.3
Unstructured	315	304	35.3	38.2	1074	979	4.5	6.3
Jointly(LSTM)	117	95	79.5	91.6	179	90	49.3	69.7
Jointly(A-LSTM)	134	123	78.6	90.9	167	73	52.9	75.5
CNN+TransE	–	–	–	–	181	91	49.6	67.4
CNNM	110	101	75.6	89.1	181	70	52.4	72.4
ACNNM	105	**94**	78.8	90.3	**166**	**68**	53.1	74.6
PACNNM	**104**	**94**	77.5	89.0	171	69	52.0	71.5

Tab. II *Results on link prediction on two datasets.*

From the results, we make the following observations. One observation is that compared with CNN+TransE, which is also based on CNN, all proposed models achieve better performance on all metrics. The reason may be that the CNN structure chosen by our models is more suitable for encoding the semantic information of descriptions. Furthermore, we introduces a gate mechanism to strengthen the semantic relation between two information sources, which is better than using only weights. The proposed models employ an attention mechanism to filter the effective information of texts, which in turn enhances the discrimination of entity representations.

As for Jointly(A-LSTM), which is the best state-of-the-art method, the performance of ACNNM is relatively competitive on both datasets, and significantly higher than baseline methods on the Mean metric. The experimental results indicate that ACNNM can effectively capture the semantic information of texts and integrate two kinds of entity representation.

Compared with the TransD model that only uses structural information, the proposed models are much better in terms of the Mean metric and are slightly worse on the Hits@10 metric. The reason is that introducing text description information can effectively alleviate data sparsity. Nevertheless, it may affect frequent entities during the training process, leading to its poorer results on the Hits@10 metric. Although our models perform worse than TransD on the Hits@10 metric, it is worth noticing that the proposed models are based on TransE rather than TransD. If we expand on other excellent representation models such as TransD, we should be able to further improve the performance of our models.

In addition, ACNNM always performs better than CNNM, which validates that the attention mechanism can enhance the semantic difference of text representation, further improving the discrimination of entity representation. On the FB15k dataset, PACNNM performs worse than ACNNM. This may be because the length of sentences on the dataset is seriously differentiated, and fixed position coding cannot effectively fit this difference. As a result, it could incur some noise into the training process. On the WN18 datasets, PACNNM and ACNNM perform similarly. The reason is that the sentences of the dataset are short and the length of sentences is close to each other. The position vector of PACNNM is more suitable for fitting the dataset.

Relation classification. To further demonstrate the performance of the models, we divides the relation into four types: 1-to-1, 1-to-N, N-to-1 and N-to-N, and compares the results of Hit@10(Filt) on FB15k under different kinds of relations.

Task	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
TransH	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransR	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
CTransR	81.5	89.0	34.7	71.2	80.8	38.6	90.1	73.8
SME(linear)	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(Bilinear)	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
Unstructured	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
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
CNNM	82.4	94.8	35.9	72.2	80.0	45.5	94.7	75.3
ACNNM	**84.9**	**95.5**	**39.6**	**74.5**	**84.6**	**49.9**	**94.9**	**77.8**
PACNNM	81.4	95.3	37.0	71.0	82.7	45.3	94.4	74.6

Tab. III *Results on the FB15k dataset by relation classification.*

Tab. III reports the Hits@10(Filt) results of the different models on these groups. From Tab. III, we can see that the improvement of ACNNM over the baseline TransE on all groups are very promising, especially in the head entity prediction under N-to-1 and N-to-N relations and the tail entity prediction under 1-to-N and N-to-N. The results demonstrate that text descriptions benefit not only simple relations, but also complex relations.

4.4 Triplet classification

Triplet classification aims to confirm whether a given triplet (h, r, t) is a correct fact or not, which is a binary classification task. In this experiment, we continue to use the two datasets FB15K and WN18 to evaluate our approach. The two test sets only contain correct triplets, which requires us to construct negative triplets. In this paper, we construct negative triplets following the same setting used in [18]. The method randomly replaces the head entity of a positive triplet to form a negative sample, and the replaced entity can only be selected from an entity set corresponding to the relation of the triplet. Therefore, the method avoids obvious unrelated triplets in the negative set, which makes the semantic difference between the negative triplet and the positive triplet smaller. As a result, it increases the difficulty of the triplet classification task.

Accuracy is used as the evaluation metric of the task. Specifically, for triplet classification, we set a threshold ϵ_r for each relation r. For a triplet (h, r, t), if its score is less than ϵ_r, it will be classified as a negative triplet, otherwise a positive triplet. The task first reaches the maximum accuracy of the validation set and obtains the threshold ϵ_r of each relationship. On the FB15k dataset, some relations appear in the validation set but not in the test set. The average value of thresholds of the relations that have occurred in the validation set is used as the missing threshold. The triplet classification accuracy on the two datasets is shown in Tab. IV. We highlight the best results in bold.

Datasets	WN18	FB15k
TransE (Baseline)	92.9	79.8
TransH	–	79.9
TransR	–	82.1
CTransR	–	84.3
TransD	–	88.0
TranSparse	–	88.5
Jointly(LSTM)	97.7	90.5
Jointly(A-LSTM)	97.8	**91.5**
CNNM	96.8	87.4
ACNNM	97.7	87.2
PACNNM	**98.1**	90.3

Tab. IV *Results on triplet classification on two datasets.*

The results reveal that our joint encoding models perform better than the baseline method TransE, and the performance of ACNNM is relatively competitive compared with the state-of-the-art methods. The results indicate that the proposed text encoding method can effectively encode semantic information, and can be well integrated into the entity representation, which enhances the semantic distinction of triplets.

Compared with CNNM and ACNNM, PACNNM achieves the best performance on both dataset. The results indicate that the PACNNM model, which introduces position information, can find more precise thresholds for complex relations. Therefore, the gap between positive and negative sample scores is enhanced. On the FB15K dataset, ACNNM is slightly worse than CNNM. However, ACNNM achieves a better performance than CNNM on the WN18 dataset. This may be because the length of descriptions in FB15k is longer than WN18, and the number of relations is much larger than WN18. The attention mechanism of ACNNM weakens the difference of scores when simulating such semantic information.

5. Conclusions and future work

This paper proposes a joint knowledge representation model based on CNN with an attention mechanism, and improves knowledge representation by introducing entity descriptions. Firstly, we proposes a text encoder based on CNN. Secondly, we design an attention mechanism to filter the text description most relevant to the relation. Furthermore, position information is introduced to extend the model. Finally, a gate mechanism is used to combine structure and textual information to obtain the final joint representation. Extensive experiments on the tasks of link prediction and triplet classification show that our models bring promising improvements to TransE. In particular, the propose model outperforms all baseline methods on the relation classification task. In the future, we will extend the model based on other excellent knowledge representation models such as TransD. Additionally, we will extend our model to encode category information, using categories as constraints on entities.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments. This work was funded by Doctoral Start-up Fund of Jianghan University and Scientific Research Planning and Guiding Program of Hubei Provincial Department of Education.

References

[1] BORDES A., GLOROT X., WESTON J. Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012, pp. 127–135.

[2] BORDES A., GLOROT X., WESTON J., BENGIO Y. A semantic matching energy function for learning with multi-relational data. *Machine Learning.* 2013, 94(2), pp. 233–259.

[3] BORDES A., USUNIER N., GARCIA-DURAN A., WESTON J., YAKHNENKO O. Translating Embeddings for Modeling Multi-relational Data. In: *Conference on Neural Information Processing Systems (NIPS)*, 2013, pp. 2787–2795.

[4] BORDES A., WESTON J., COLLOBERT R., BENGIO Y. Learning Structured Embeddings of Knowledge Bases. In: *AAAI Conference on Artificial Intelligence (AAAI)*, 2011, pp. 301–306.

[5] CHANG G., HUO H. A method of fine-grained short text sentiment analysis based on machine learning. *Neural Network World*. 2018, 28(4), pp. 325–344.

[6] CHANG K.-W., YIH W.-t., YANG B., MEEK C. Typed Tensor Decomposition of Knowledge Bases for Relation Extraction. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1568–1579.

[7] DAS R., NEELAKANTAN A., BELANGER D., MCCALLUM A. Chains of Reasoning over Entities, Relations, and Text using Recurrent Neural Networks. In: *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017, pp. 132–141.

[8] FAN M., ZHOU Q., ZHENG T.F., GRISHMAN R. Distributed Representation Learning for Knowledge Graphs with Entity Descriptions. *Pattern Recognition Letters*. 2016, 93, pp. 31–37.

[9] GAO W., PENG M., WANG H., ZHANG Y., HAN W., HU G., XIE Q. Generation of topic evolution graphs from short text streams. *Neurocomputing*. 2020, 383(28), pp. 282–294.

[10] GAO W., PENG M., WANG H., ZHANG Y., XIE Q., TIAN G. Incorporating word embeddings into topic modeling of short text. *Knowledge and Information Systems*. 2019, 61(2), pp. 1123–1145.

[11] JI G., HE S., XU L., LIU K., ZHAO J. Knowledge Graph Embedding via Dynamic Mapping Matrix. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015, pp. 687–696.

[12] JI G., LIU K., HE S., ZHAO J. Knowledge graph completion with adaptive sparse transfer matrix. In: *AAAI Conference on Artificial Intelligence (AAAI)*, 2016, pp. 985–991.

[13] KIM Y. Convolutional Neural Networks for Sentence Classification. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[14] LIN Y., LIU Z., SUN M., LIU Y., ZHU X. Learning entity and relation embeddings for knowledge graph completion. In: *AAAI Conference on Artificial Intelligence (AAAI)*, 2015, pp. 2181–2187.

[15] LIN Y., HAN X., XIE R., LIU Z., SUN M. Knowledge representation learning: A quantitative review. *arXiv preprint arXiv:1812.10901*. 2018.

[16] LIN LI Kefeng Fan Z.Z., XIA Z. Community detection algorithm based on local expansion k-means. *Neural Network World*. 2016, 26(6), pp. 589–605.

[17] MIKOLOV T., YIH W.T., ZWEIG G. Linguistic regularities in continuous space word representations. In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2013, pp. 746–751.

[18] SOCHER R., CHEN D., MANNING C.D., NG A.Y. Reasoning with neural tensor networks for knowledge base completion. In: *Conference on Neural Information Processing Systems (NIPS)*, 2013, pp. 926–934.

[19] SRIVASTAVA N., HINTON G., KRIZHEVSKY A., SUTSKEVER I., SALAKHUTDINOV R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 2014, 15(1), pp. 1929–1958.

[20] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A.N., KAISER L., POLOSUKHIN I. Attention Is All You Need. In: *Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.

[21] WANG Z., ZHANG J., FENG J., CHEN Z. Knowledge Graph and Text Jointly Embedding. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1591–1601.

[22] WANG Z., ZHANG J., FENG J., CHEN Z. Knowledge graph embedding by translating on hyperplanes. In: *AAAI Conference on Artificial Intelligence (AAAI)*, 2014, pp. 1112–1119.

[23] XIE R., LIU Z., JIA J., LUAN H., SUN M. Representation Learning of Knowledge Graphs with Entity Descriptions. In: *AAAI Conference on Artificial Intelligence (AAAI)*, 2016, pp. 2659–2665.

[24] XU J., QIU X., CHEN K., HUANG X. Knowledge Graph Representation with Jointly Structural and Textual Encoding. In: *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 1318–1324.

[25] YANG F., YANG Z., COHEN W.W. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In: *Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 2319–2328.

[26] YANKAI LIN Haozhe Ji Z.L., SUN M. Denoising Distantly Supervised Open-Domain Question Answering. In: *Conference on the 56th annual meeting of the association for computational linguistics (ACL)*, 2018, pp. 1736–1745.

[27] ZHANG D., YUAN B., WANG D., LIU R. Joint Semantic Relevance Learning with Text Data and Graph Knowledge. In: *Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, 2015, pp. 32–40.

[28] ZHONG H., ZHANG J., WANG Z., WAN H., CHEN Z. Aligning Knowledge and Text Embeddings by Entity Descriptions. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 267–272.