BCRL: Long Text Friendly Knowledge Graph Representation Learning

Gang Wu[1,2(✉)], Wenfang Wu[1], Leilei Li[1], Guodong Zhao[1], Donghong Han[1,2], and Baiyou Qiao[1,2]

[1] School of Computer Science and Engineering, Northeastern University, Shenyang, China
wugang@mail.neu.edu.cn, 389446497@qq.com, leilei-li@foxmail.com, gdzhao@stumail.neu.edu.cn, {handonghong,qiaobaiyou}@mail.neu.edu.cn

[2] Key Laboratory of Intelligent Computing in Medical Image, Ministry of Education, Northeastern University, Shenyang, China

Abstract. The sparse data and large computational overhead in the use of large-scale knowledge graphs have caused widespread attention to Knowledge Representation Learning (KRL) technology. Although many KRL models have been proposed to embed structure information, their ability to accurately represent newly added entities or entities with few relations is significantly insufficient. In some studies, the introduction of textual information has partially solved this problem. However, most existing text-enhanced models only consider the shallow description information of the entities, and ignore the relation mention information between entities, and deep semantic information between sentences and words, which is not optimized for long texts supplementary information like Wikipedia.

In this paper, we proposed a long text friendly structure-text joint KRL model, named BCRL (BERT and CNN Representation Learning), which can effectively explore rich semantics embedded in entity description and relation mention text taking Wikipedia as supplementary information. For the obtained text of entity description and relation mention, the model first uses the BERT model to generate sentence vector representation respectively. Then it uses a convolutional neural network with an attention mechanism to select valid information in the text and obtain the overall vector representation of the text. Finally, the gate mechanism is used to combine the structure-based and the text-based vectors to generate the final joint representation. We evaluated the performance of our BCRL model on link prediction tasks using FB15K and WN18 datasets. The experimental results show that BCRL outperforms structure-only models and text-enhanced models in most cases, and has significant advantages in complex relation representation.

Keywords: Knowledge Representation Learning · Long text · BERT · Convolutional neural network

Supported by the National Key R&D Program of China (Grant No. 2019YFB1405302) and the NSFC (Grant No. 61872072 and No. 61672144).

J. Z. Pan et al. (Eds.): ISWC 2020, LNCS 12506, pp. 636–653, 2020.
https://doi.org/10.1007/978-3-030-62419-4_36

1 Introduction

In recent years, the Knowledge Graph (KG) has received extensive attention from academia and industry for its powerful semantic expression capabilities, and has been widely used in fields such as question answering [5,21] systems and web search. In order to solve the problems of low computing efficiency and sparse data, KRL technology has been widely concerned. Its main goal is to represent the entities and relations of a KG in a low-dimensional dense real-valued vector space. In this way, it improves the efficiency of complex semantic relationship computation within entities, relations, and between them.

The translation models typified by TransE (Translating Embedding) [3] are recent research hotspots of KRL. They are not only simple in model, high in computational efficiency, but also can guarantee good knowledge expression ability. However, their ability to accurately represent newly added entities or entities with few relationships is insufficient because only the structural information of triples is taken into consideration in such models. To tackle this problem, some work began to introduce textual information [1,16] to help improve knowledge representation. Entity description is the most common type of such textual information. As exemplified in Fig. 1, the head entity and the tail entity of a triple from the Freebase KG are each associated with a piece of textual description.

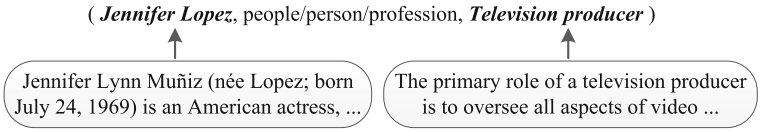

Fig. 1. An example of entity descriptions in Freebase

However, the existing text-enhanced KRL methods are still facing challenges.

i) It is difficult to capture the exact meaning of relations in context text. A typical situation is how to distinguish multiple different semantics of the same relation. For example, the relation "parentOf" can mean either "being the father of" or "being the mother of" depending on the entities in triples.

ii) The representation of entity description is not comprehensive enough. For example, the semantics between sentences (or words) in the case of long description texts are usually ignored. And the reflection of the semantic difference of the same entity in different triple contexts is generally lacking.

One of the possible reasons behind these problems is that existing methods do not make full use of the rich semantics in long texts, i.e. multiple sentences. Therefore, we proposed a long text friendly structure-text joint KRL model, named BCRL, which can effectively explore rich semantics embedded in entity description and relation mention texts that are obtained from Wikipedia as supplementary information. Firstly, the model obtains accurate text information of entities and relations through lemmatization, stop words removal, and similarity

calculations for relation mentions. Secondly, the BERT model is used to obtain the sentence vector and learn the semantic information between words in the sentence. Furthermore, the CNN with sentence-level positional information coding is employed to learn the semantic information between sentences to obtain the overall vector representation of the text. Finally, the gate mechanism is introduced to realize the joint representation of structural information and textual information on top of the TransE framework. In addition, for the entity description text, a relation-related attention mechanism is added to further enhance the text embedding of the entity.

In summary, the contributions of this paper are as follows:

1. The proposed model achieves long text friendly by introducing BERT and CNN to gradually capture the semantics of different granularities (word level and sentence level) in the text.
2. To meet the first challenge, relation mention information is introduced in the model to enhance the knowledge representation as well, which is obtained from all entity descriptions in the triples involved by the relation.
3. To meet the second challenge, the model makes the representation of entity description more comprehensive by introducing a relation-oriented attention mechanism that captures the most relevant information in the entity description in different contexts through a triple-relation text vector.

We evaluate our model on link prediction task, using benchmark datasets from Freebase and Wordnet with the text corpus. Experimental results show that, our model achieves the state-of-the-art performance, and significantly outperforms previous text-enhanced models.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the long text friendly text representation model in detail. Section 4 presents the structure-text joint learning. Empirical evaluation of the proposed model and comparison with other state-of-the art models are presented in Sect. 5. Finally, Sect. 6 summarises the whole paper and points out some future work.

2 Related Work

2.1 Translation-Based Models

In recent years, there has been a great deal of work on KRL, and most studies concentrate on translation-based models. This kind of models propose to embed both entities and relations into a continuous low-dimensional vector space according to some distance-based scoring functions.

One of the most representative translation models is TransE which regards the relationships in the KG as some kind of translation vector between entities. Specifically, for each fact triple (h, r, t), it represents entities and relationships in the same vector space, and considers the relationship vector r as the translation between the head entity vector h and the tail entity vector t, *i.e.*, "$h + r \approx t$".

Thus, the scoring function is defined as $f_r(h,t) = \|h + r - t\|_{L1/L2}$. Where h, r and t represent the vectors of head entity h, relation r and tail entity t, respectively. And $L1$ and $L2$ represent 1-norm and 2-norm respectively. If the fact (h,r,t) is true, the score $f_r(h,t)$ tends to be close to zero.

Though TransE is an effective KRL model for representing 1-to-1 relation, its rough translation idea has flaws in dealing with more complicated relations like 1-to-N, N-to-1 and N-to-N. This motivates the proposal of subsequent improvements such as TransH [17], TransR [9], TransD [8], etc., which allow entities to have different representations when different relationships are involved. TransE is a simple and efficient method for KRL.

2.2 Introducing Text Information

In order to improve the KRL, many research works have been proposed to embed text information to improve the knowledge representation.

Embedding KGs with textual information to improve the knowledge representation can be traced back to the neural tensor network model (NTN) proposed by Socher et al. [13], where textual information is simply used to initialize the entity representation. Specifically, NTN first learns word embeddings from the auxiliary news corpus, and then initializes the representation of each entity by averaging the vectors of words contained in its name. For example, the embedding of AlfredHitchcock is initialized by the average word embeddings of Alfred and Hitchcock. Since this method separates text information from KG facts, it cannot effectively utilize the interactive information between fact triple entities. Moreover, the method initializes the representation only on the basis of the entity name, which makes it impossible to make full use of textual information.

Wang et al. [17] proposed a joint model, which aligns the entity name and the Wikipedia anchor text to project KG's knowledge and Wikipedia text into the same space, which can better use text information in the embedding process and improve the accuracy of fact prediction.

Toutanova et al. [14] used convolutional neural networks to derive continuous representations for text relations, which has greatly improved entities with text representations. Xu et al. [19] learns the joint representation of structure and text through LSTM network with gate mechanism.

Xie et al. [18] proposed a text-enhanced TransE model which uses continuous bag-of-words model and CNN to encode the entity description information. The model jointly represents the structure-based and description-based two parts. The former captures the structural information of the facts of the KGs, and the latter captures the textual information of the entity description. Although CNN is used to encode text information, it only includes convolutional layers and pooling layers, and cannot learn semantic information between multiple sentences of text. In addition, the method has not considered the filtering and screening of textual information and the effective form of joint representation.

3 Long Text Friendly Text Representation

As stated in the introduction, effective use of long texts may be one of the possible ways to better meet the novel challenges of KRL. In this section, we first present the preparation of long texts for entities and relations respectively, and then focus on the long text friendly text representation model.

3.1 Long Text Friendly Text Information Extraction

Entity Description Extraction. The sparseness and staleness of entity description information is very common in a single KG. Taking Freebase as an example, due to the premature establishment of the knowledge base, a lot of information is out of date, and the length of different entity description varies widely, from 350 words to several words. Linking Wikipedia information for KG entities is a commonly used solution for this case. For example, Freebase provides entity mapping files on Wikipedia[1]. In this paper, the abstract text of the linked Wikipedia entry is taken as the supplementary of the entity description in this way. General entity link tools can also be used to obtain the corresponding supplementary information, such as TAGME [6] and AIDA [20].

Relation Mention Extraction. For the relation in a triple, it is acceptable to supplement text information with the entities mentioned in the triple. To this end, a corpus is built with all the text corresponding to the Wikipedia entries linked in the entity linking process. Then the text of a relation mention can be extracted from the corpus. The relation dataset is made available on GitHub[2]. Specifically, given a relation r of the triple (h, r, t), all sentences containing both the head entity h and the tail entity t in the triple are extracted from the corpus as candidate relation mentions [1].

Obviously, this will involve a lot of noise that is not actually related to the relation r, which will affect subsequent textual representations. In order to effectively filter noise, similarity calculations are performed between relations and their candidate mentions from the lexical level and the semantic level respectively.

For the lexical-level, a candidate mention sentence s is determined to be similar to the relation r if any of the synonyms and superordinate words of r in WordNet are found in s. For example, for a triple (Pain,/medicine/disease/prevention_factors, Capsicum), a sentence can be regarded as an accurate relation only if the sentence contains the triple head and tail entities and at least one synonym or superordinate word about the relation Mention. In the example, *medicine* and *disease* are the hypernyms of $prevention\_factors$, *drug* is a synonym of *medicine* in WordNet, and the relationship set is (prevention_factors, disease, medicine, drug).

[1] http://storage.googleapis.com/freebase-public/fb2w.nt.gz.
[2] https://github.com/BoBoManTou/KG.

Hence, its similarity can be calculated by the vector space model. For example, the sentence "Many capsicum medicines have been used in the management of pain in various traditional systems" is processed into text to obtain a set of words medicine, manage, tradition, system. Each word represents a dimension, and its value is 0 or 1, indicating whether the word appears in the current text. Take {prevention_factors, disease, medicine, drug, manage, tradition, system} as the dimension to get mention and relationship. The two vector representations are $0, 0, 1, 0, 1, 1, 1, 1$ and $1, 1, 1, 1, 0, 0, 0$. Suppose m represents the candidate relation mention set, r is the corresponding relation set, V_m represents the space vector representation of the mention set, and V_r represents the vector representation of the relation set. Then the similarity between the two can be expressed by the cosine distance. The calculation method is shown in Eq. 1.

$$\cos\left(V_m, V_r\right) = \frac{V_m \cdot V_r}{|V_m||V_r|} \tag{1}$$

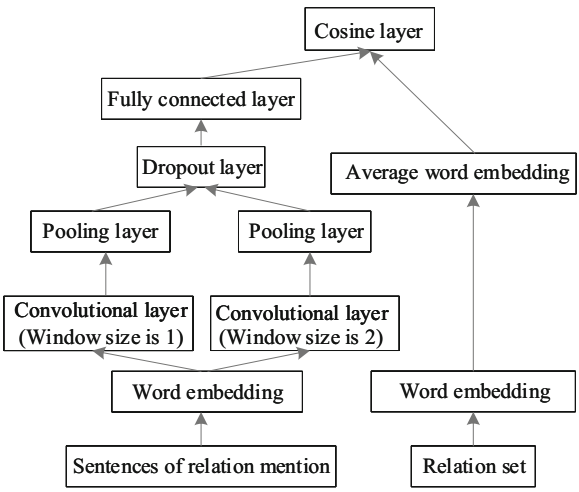

Fig. 2. Semantic level similarity calculation

Further filtering from the semantic level similarity is necessary especially when the relation mention sentence does not contain any superordinate words or synonyms of the corresponding relation. Here, a combination of CNN and Skip-gram [11] is developed to model candidate relation mention sentences in semantic vectors, and the vector space model can be used to calculate the similarity to the word embedding of the relation. As shown in Fig. 2, two parallel CNN models are used to learn the vector representation of the sentence mentioned in the candidate relations, and the average word embedding method is used to learn the vector representation of the relation.

For relation-mentioned sentences, the beginning of the model is to use the Skip-gram [11] model to obtain the word embedding of the relation-mentioned sentences based on the corpus in the previous article. Two convolution kernels with different window sizes of 1 and 2 are used in the convolution layer to extract local features with different granularities to maximize information utilization. In this paper, the activation function in the convolutional layer uses ReLU. The pooling layer after the convolutional layer is used to select a variety of semantic combinations, extract the main features, and change the variable-length input into a fixed-length output. The pooling layer adopts Max-pooling operation, and selects the strongest value of the input vector in each window to form a new vector. The output after the pooling operation passes through a Dropout layer. Dropout sets each feature extracted by the pooling layer to 0 with a certain probability. This can avoid overfitting caused by the model's excessive dependence on certain features, thereby improving the generalization ability of the model. For the extracted main features, the non-linear recombination is performed through the fully connected layer to obtain the semantic vector representation of the input mentioned sentence.

For the relation set, the model also uses the skip-gram model to obtain the word embedding of the relation set. Then this paper obtains the vector representation of the relation set by averaging the word embeddings of all the words in the set. Finally, the cosine distance is used to express the semantic similarity between the relation mention sentence and the corresponding relation set. Suppose m represents the candidate relation mention sentence, r is the corresponding relation set, V_m represents the semantic vector representation of the reference set, and V_r represents the semantic vector representation of the relation set, so the similarity between the two can be calculated by Eq. 2. If the similarity exceeds the set threshold ε, the sentence is mentioned as the exact text of the relationship.

$$sim(m, r) = \cos(V_m, V_r) \tag{2}$$

3.2 Text Representation Model of BCRL

In the design of the text representation model of BCRL, several technologies are introduced and integrated to adapt to long texts, including BERT, CNN, attention mechanism, and sentence position coding. Figure 3 shows the overall framework of the text-enhanced representation model. The function of each part of the model is explained in detail below.

The Overall Framework of Text Representation Model

The above entity description and relation mention extraction methods bring more accurate long text information. In order to embed as much semantic information as possible within and between sentences into the text representation, we propose to combine the BERT language model and the CNN. Sentence sequence vectors are first generated by the BERT model, and then these sentence-level feature vectors are input into a convolutional neural network to form final overall text vector. In addition, the attention mechanism and position coding are added to CNN to further enrich textual representation of the entity description.

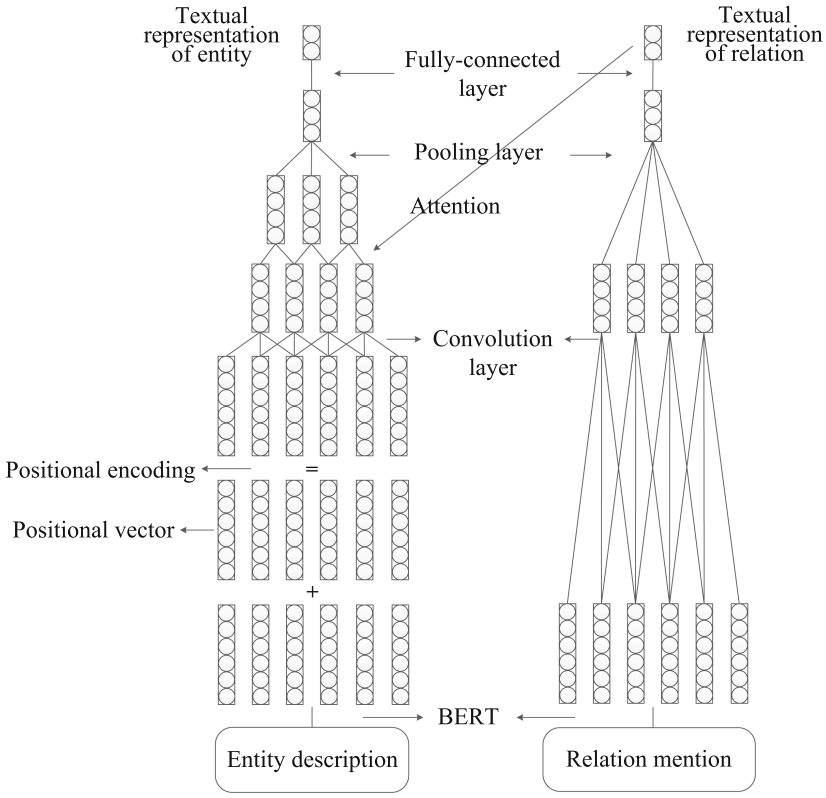

Fig. 3. The overall framework of text representation model

BERT. The BERT model is derived from the paper [4], which is a language model based on two-way Transformer proposed by Google. This paper uses the BERT model to obtain the sentence vector representation of the text. In order to achieve this, the value of BERT's parameter *max_seq_len* should be increased. Hence, we set the value of max_seq_len to be the average length of the entity description texts. Although theoretically the output value of any layer of the transformer can be used as a sentence vector, the experimental results show that the value of the penultimate layer is better. The input of the BERT model is a sequence of preprocessed sentences d where the sequence length is n and each sentence contains m words. Thus, the input is defined as $d_1 : n = d_1, d_2, \cdots, d_n$. Where $d_i \in D^m$ representing m words of the ith sentence of an entity description text. For the sentence sequence d, in order to prevent overfitting problems in text processing, the output value of the penultimate layer with a dimension of 768 is selected as the output sentence vector v.

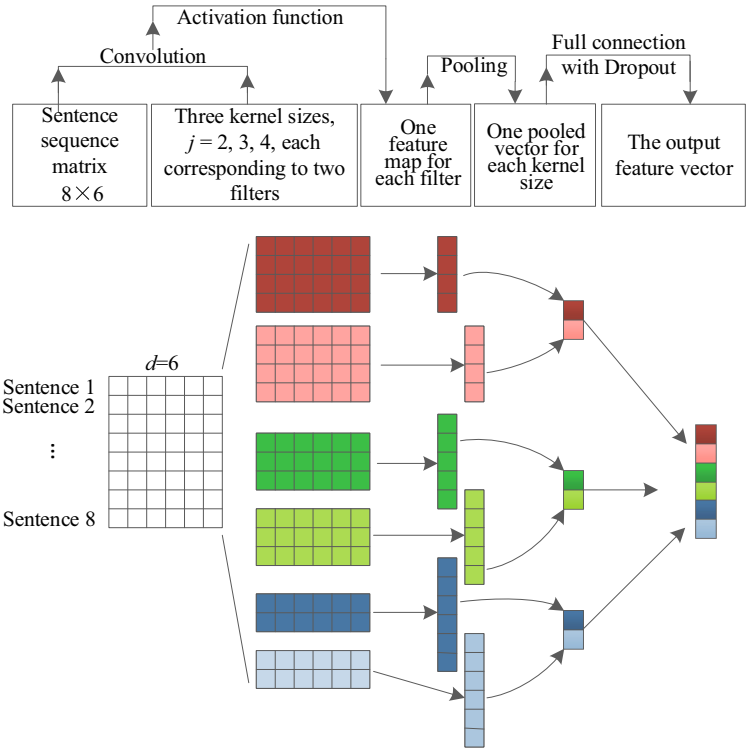

Fig. 4. The CNN network

CNN. The CNN network consists of a convolutional layer, a pooling layer, a Dropout layer, and a fully connected layer. Figure 4 illustrates an example that takes a sequence of 8 sentence vectors as input. Each sentence dimension is 6. For each convolution kernel size $j = 2$, 3, and 4, the convolution operation, pooling operation, and full connection are performed in sequence.

Specifically, in our model, the input of the CNN convolutional layer are n sentence vectors v obtained by previous BERT where each sentence dimension is 768. The convolution layer performs convolution operation on these n sentence vectors with a sliding window of size j, and outputs the feature map q. The sentence vector sequence processed by the sliding window is defined as $v_{i:i+j-1} = v_i, v_{i+1}, \ldots, v_{i+j-1}$. The i-th output feature vector after convolution is shown in Eq. 3, where $w \in R^{j \times m}$ is the filter, $b \in R$ is the bias term, and f is the activation function. In this paper, RELU is selected as the activation function.

$$q_i = f(w \cdot v_{i,i+j-1} + b) \tag{3}$$

The first k maximum pooling (K-Max Pooling) is used here, i.e., the first k maximum values are selected for the input vectors in each window to form a new vector. The i-th vector output by the pooling layer with a window size of

n_p can be calculated by using Eq. 4. When the number of filters is l, the output of the pooling layer is $p = [p_1, \ldots, p_l]$.

$$p_i = \max_k(q_{n_p \cdot i}, \ldots, q_{n_p \cdot (i+1)-1}) \tag{4}$$

The model also provides a Dropout layer working in a Bernoulli process to further prevent overfitting. As shown in Eq. 5, the output of the Dropout layer is $\tilde{p}$ where vector $\hat{\beta} \sim \text{Bernoulli}(\rho)$ with probability ρ.

$$\tilde{p} = \hat{\beta} * p \tag{5}$$

The output of the CNN fully connected layer is defined as Eq. 6 where w_o is a parameter matrix and b_o is an optional bias term.

$$e_d = w_o \cdot \tilde{p} + b_o \tag{6}$$

Relation-Based Attention for Entity Text Representation
As we know, in addition to the common 1-1 relations, there are also complex relations such as 1-N, N-1, and N-N in a knowledge graph. For the entity description information, CNN semantically encodes the entire text, without considering that the description information contains the different semantics of the entity under multiple relations. This means that given a triple, the relationship is given, which will cause some interference for the entity description to contain information about other relations. In order to make CNN sensitive to the different semantics of the entity under various relations, an attention mechanism is integrated between the convolutional layer and the pooling layer. In this way, the most relevant relations between entities can be effectively captured with the help of generated relation mention vectors.

Given the sentence vector sequence $v_{1:n} = v_1, v_2, \ldots, v_n$ of an entity and a relation $r \in R^m$, the textual representation r_d of the relation r is believed to be closely related to the relationship mention information. Therefore, the relation-based attention of entity description is defined as Eq. 7. Suppose the output of the convolution layer is q, then the output with the relation-based attention is defined to be $\tilde{q} = q\alpha(r)$, which can be used as the input of the pooling layer.

$$\alpha(r) = Softmax(v_{1:n} r_d) \tag{7}$$

Sentence Level Positional Encoding for Entity Text Representation
Another useful but overlooked feature is the sentence order information in the sentence sequence. Although BERT considers the order information of the words in the sentence, CNN does not include the order feature of the sentence when encoding the whole text, and part of the semantics may be lost. Therefore, to make effective use of this information, the sentence position is encoded as position vector γ_i and then combined with the sentence vector v_i into a new vector C_i by addition.

The position vector γ_i is generated by using the sine and cosine functions on position *pos* at different frequencies according to Vaswani's method, expressed as

Eqs. 8 and 9. Here, pos corresponds to the input position, and d is the dimension of the position vector.

$$\gamma_{(pos,2i)} = \sin pos/10000^{2i/d} \tag{8}$$

$$\gamma_{(pos,2i+1)} = \cos pos/10000^{2i/d} \tag{9}$$

Given a sentence sequence vector $v_{1:n} = v_1, \cdots, v_n$, its position vector $\gamma_{1:n} = (\gamma_1, \cdots, \gamma_n)$, the new input of CNN after adding location information is $C_{1:n} = (v_1 + \gamma_1, \cdots, v_n + \gamma_n)$.

4 Structure-Text Joint Knowledge Graph Learning

4.1 TransE-Based Structural Representation

TransE-based representation models perform well in tasks such as knowledge reasoning and relationship extraction, and have become a research hotspot for knowledge representation.

Given a triple (head entity, relation, tail entity), express it as (h, r, t). The corresponding vector of the triple (h, r, t) is represented as $(\mathbf{h}, \mathbf{r}, \mathbf{t})$. TransE aims to express entities and relationships as low-dimensional continuous vectors. The legal triple vector should satisfy the formula $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, and the wrong triple does not. Therefore, TransE defines the following score function to measure the quality of the triple, as shown in Eq. 10.

$$\begin{aligned} f_r(h,t) &= \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_1/L_2}, \\ s.t. &: \|\mathbf{h}\|_2^2 \leq 1; \|\mathbf{t}\|_2^2 \leq 1 \end{aligned} \tag{10}$$

Equation 10 is the L1 or L2 distance between vectors $\mathbf{h}+\mathbf{r}$ and $\mathbf{t}$. For a reasonable scoring function, the score of the legal triple is lower than the score of the wrong triple.

4.2 Structure-Text Joint Representation

As shown in Fig. 5, the model jointly expresses structure information and text information through a gate mechanism.

In this paper, the gate mechanism proposed by Xu et al. [19] is used for the fusion of the learned textual representation and the structural representation from TransE. As defined in Eq. 11 and 12, it means that the joint representation V_j is regarded as the result of the weighted sum of the textual representation V_d and the structural representation V_s. Here, g_s and g_d are the gates that balance the two information sources, and $\odot$ is the element multiplication.

$$V_j = g_s \odot V_s + g_d \odot V_d \tag{11}$$

$$s.t.\ g_d = 1 - g_s; g_s, g_d \in [0, 1] \tag{12}$$

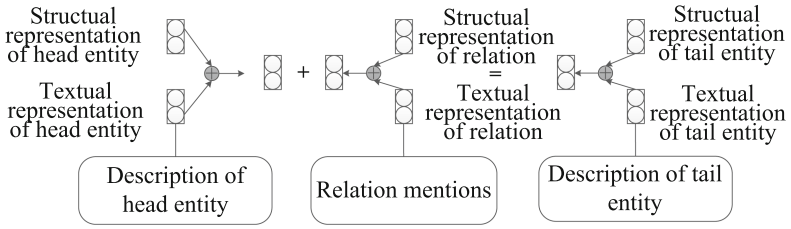

Fig. 5. Structure-text joint representation

The gate g is defined as $g = Softmax(\hat{g})$, where $\hat{g} \sim \text{uniform}(0, 1)$ is a real-valued vector initialized randomly in a uniform distribution. Softmax function is employed here to constrain the value of the gate control to [0, 1]. Note that the Sigmoid function is also applicable for computing the gate as stated in [19].

Similar to the TransE series model, the structure-text joint representation score function is defined as shown in Eq. 13.

$$f(h, r, t; d_h, d_r, d_t) = ||(g_{hs} \odot h_s + g_{hd} \odot h_d) + (g_{rs} \odot r_s + g_{rd} \odot r_d) - (g_{ts} \odot t_s + g_{td} \odot t_d)||_{L1/L2} \tag{13}$$

Among them, g_{hs} and g_{hd} are the doors of the head entity, g_{rs} and g_{rd} are the door of the relation, and g_{ts} and g_{td} are the door of the tail entity.

4.3 Model Training

According to the TransE, the maximum interval method [3] is utilized to train the model. The loss function of the triples (h, r, t) is described in Eq. 14 where f is the score function of our model, $\gamma > 0$ is the margin between golden tuples and negative tuples, D is the set of valid triples in the KG, and $\hat{D}$ is the set of invalid triples not in the KG.

$$L = \sum_{(h,r,t)\in D} \sum_{(\hat{h},\hat{r},\hat{t})\in \hat{D}} \left[f\left(h, r, t\right) + \gamma - f\left(\hat{h}, \hat{r}, \hat{t}\right) \right]_+ \tag{14}$$

This paper uses the method proposed by Wang et al. [17] to set different probabilities to replace the head or tail entities according to the Bernoulli distribution, which divides the relations into four different types according to the number of connected entities at both ends: 1-1, 1-N, N-1, and N-N. If it is a 1-N relation, it increases the chance of replacing the head entity, and if it is an N-1 relation, it increases the chance of replacing the tail entity, which can effectively improve the model training effect. For each triple, a valid triple (h, r, t) is defined as $\hat{D} = \left\{ \left(\hat{h}, r, t\right) \right\} \cup \{(h, \hat{r}, t)\} \cup \left\{ \left(h, r, \hat{t}\right) \right\}$.

5 Experiments

5.1 Experiment Settings

Two popular KRL benchmark datasets FB15K and WN18[3] are chosen in the experiments. The inverse relations of the existing relations are considered to expand the datasets. In this way, the number of relations and training triples are doubled.

Initial entity descriptions are available from GitHub[4]. A simplified corpus was then built on the existing entities by performing entity linking with the English Wikipedia[5] data (May 16, 2019) which is about 15.5G in size and contains more than 1.2 billion words. The entity description representation and relation mention representation in the experiments are all based on this corpus.

In order to accelerate the convergence, the vectors and matrices of BCRL are initialized through the RTransE [7] model. The entity/relation vector dimension $d \in \{50, 100\}$, the learning rate $\alpha \in \{0.01, 0.001, 0.0001, 0.0005\}$, and the maximum interval $\gamma \in \{0.1, 1, 2, 4, 4.5, 5, 5.5, 6\}$. The pre-trained BERT in the text representation model is BERT-Base-Uncased. The window size of the convolutional layer $j \in \{2, 3, 4, 5\}$, the number of filters $l \in \{50, 100\}$, and the drop rate is set to 0.5. The L1 normal form is used in the scoring function. The training process iterates the MBGD (mini-batch gradient descent method) algorithm 2000 times.

In order to better compare with other knowledge representation learning models such as TransE, the same evaluation criteria as TransE are used, i.e., Mean Rank and Hits@10. The smaller the Mean Rank is, the better the Hits@10 is.

In addition, in order to better analyze the impact of text information on knowledge graph representation learning, we divide the relations into four types: 1-1, 1-N, N-1, and N-N. Compare the results of Hits@10 (Filtered) on the dataset.

5.2 Experiment Introduction

Link prediction refers to the task of predicting entities that may have specific relations with a given entity. Specifically, for a triple (h, r, t), it means to predict tail t when given head h and relation r, and to predict head h when given relation r and tail t. The former can be denoted as $(?, r, t)$, and the latter can be denoted as $(h, r, ?)$. The candidate prediction result entities are returned as a ranked set.

Two sets of comparative experiments were performed on the link prediction task to evaluate the performance of the proposed BCRL model. The comparison models can be divided into two categories.

– Structure-only models: SME [2], TransE [3], TransH [17], TransR [9], TransD [8], HolE [12], ANALOGY [10], CompleEx [15].

[3] https://everest.hds.utc.fr/doku.php?id=en:transe.
[4] https://github.com/xrb92/DKRL.
[5] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2.

– Text-enhanced model: Jointly(LSTM) [19], Jointly(A-LSTM) [19], TEKE_E [1], AATE_E [1], CNN+TransE [18].

Experiment 1. Compare the BCRL model with other representation learning models such as that those use only structural information and that those introduce textual information, and evaluate the relative accuracy of the model in the average ranking and the top ten rankings. In order to investigate the effect of attention mechanism and position information introduced in BCRL on the ability of model representation, BCRL-A (add attention mechanism), BCRL-PA (add location and attention mechanism) and BCRL (neither of them are added) were also added for comparative experiments.

Experiment 2. In order to specifically analyze the impact of text information on different relations, BCRL, BCRL-A, BCRL-PA, TransE, and other representation learning models that introduce textual information were compared experimentally for 1-1, 1-N, N-1 and N-N four relations. This part of the experiment was only done on the FB15K data set.

5.3 Experiment Results

Table 1. The result of the experiment about BCRL on the task of link prediction.

Method	WN18				FB15K			
	MeanRank		Hits@10		MeanRank		Hits@10	
	Raw	Filt	Raw	Filt	Raw	Filt	Raw	Filt
SME	545	533	65.1	74.1	274	154	30.7	40.8
TransE	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
TransD	224	212	79.6	92.2	194	91	53.4	77.3
HolE	–	–	–	**94.9**	-	65	–	81.0
ANALOGY	–	–	–	94.7	–	–	–	**85.4**
CompleEx	–	–	–	94.7	–	–	–	84.0
CNN+TransE	–	–	–	–	181	91	49.6	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	73	52.9	75.5
TEKE_E	–	127	–	93.8	–	79	-	67.6
AATE_E	–	123	–	94.1	–	76	-	76.1
TransE (our)	304	291	72.4	82.5	211	75	49.1	65.0
BCRL	110	97	77.7	92.3	165	67	53.6	83.5
BCRL-A	107	92	78.7	94.5	**159**	**63**	**55.3**	84.7
BCRL-PA	**106**	**90**	**80.7**	**94.9**	164	67	52.9	82.3

Experiment 1. In order to exhibit the performance under the same environment, we implemented both the TransE model and the BCRL model. The performance of our TransE is significantly different from that of the original paper TransE system on the FB15K dataset. All the results of Experiment 1 are listed in Table 1. The best result values in each group of experiments are highlighted in bold, and the underlined ones indicate the suboptimal values. The result values of the baseline evaluations are from their original work. The "–"s in the table indicate those results not reported in previous work. The same applies to the following experimental result table.

The following conclusions can be drawn according to Table 1.

- The performance of the BCRL-A model in this paper is significantly better than the TransE model (TransE is a baseline KRL model). For the WN18 and FB15K datasets, the average ranking effect has improved by 64.8%, 68.4%, 8.7%, and 14.5%, and the top ten rankings have increased by 24.6%, 16.0%, 12.6%, and 30.3%. They are also superior to the other structure-based models TransH, TransR, and TransD. The results confirm that textual information is beneficial to a structure-based knowledge graph representation learning model.
- The metrics of the BCRL-A model on the WN18 and FB15K datasets are similar to those of the current best semantic matching model ANALOGY, and the MeanRank on the FB15k dataset has achieved the best results so far. Since our BCRL model is simply based on the TransE framework, there is still much room for improvement.
- Compared with the typical text representation model Jointly (A-LSTM), most of the metrics value of the BCRL-A model are superior, which indicates that our BCRL model can effectively capture the semantics in textual information, and has certain effects in joint representation of textual information and structural information.
- Comparing three variants of our model, BCRL-A is significantly better than BCRL, which means that introducing relation-based attention mechanism can strengthen the semantic difference of entity description information and further improve the discrimination of entity representation. On the FB15k dataset, BCRL-PA with additional position-coding information performs worse than BCRL-A with only the attention mechanism. This may be due to the differences in the number of description sentences and the length between different entities in the dataset. Position-coded information cannot effectively reflect such difference, and even becomes interference information. However, BCRL-PA performs better than BCRL-A on the WN18 dataset. A possible reason is that the sentence length of the WN18 dataset is short and the difference in sentence length is not large. Thus, the position coding is more suitable in this case.

Table 2. Hit@10 of link prediction on different type of relations on FB15k dataset.

Task	Head entity prediction				Tail entity prediction			
Relationship type	1-1	1-N	N-1	N-N	1-1	1-N	N-1	N-N
Jointly(A-LSTM)	83.8	95.1	21.1	47.9	83.0	30.8	94.7	53.1
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
BCRL	85.9	95.2	36.9	81.8	85.1	46.5	95.6	84.3
BCRL-A	**87.8**	**96.9**	**40.7**	**83.5**	**87.8**	**50.4**	**95.7**	**85.6**
BCRL-PA	85.4	95.7	37.8	81.2	85.8	46.7	95.3	84.4

Experiment 2. From Table 2, we can see that our BCRL model has better performance than the basic model on all types of relations (1-1, 1-N, N-1, and N-N). In addition, the BCRL-A model has better results than the Jointly (A-LSTM) model, especially for the head entity prediction under the N-1, N-N relation and the tail entity prediction under 1-N, N-N. Since BCRL-A and Joint (A-LSTM) are both based on TransE, we conclude that the introduction of relation mention text is very meaningful for improving overall knowledge representation.

6 Conclusions

In this paper, we propose a text-enhanced knowledge graph representation model, named BCRL, which utilizes entity description and relation mention to enhance the knowledge representations of a triple. It tackles the challenges of incomprehensive entity description representation, and inaccurate relation mention representation from the perspective of text-sentence representation. The experimental results show that BCRL can capture the semantic information of text more effectively than the previous textual information based model, and has significant improvements on the link prediction task compared with the baseline systems.

References

1. An, B., Chen, B., Han, X., Sun, L.: Accurate text-enhanced knowledge graph representation learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 745–755, June 2018
2. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data. Mach. Learn. **94**(2), 233–259 (2013). https://doi.org/10.1007/s10994-013-5363-6
3. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS 2013, pp. 2787–2795 (2013)

4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Dubey, M., Banerjee, D., Abdelkawi, A., Lehmann, J.: LC-QuAD 2.0: a large dataset for complex question answering over Wikidata and DBpedia. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11779, pp. 69–78. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30796-7_5
6. Ferragina, P., Scaiella, U.: TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM 2010, pp. 1625–1628 (2010)
7. García-Durán, A., Bordes, A., Usunier, N.: Composing relationships with translations. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 286–290, September 2015
8. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 687–696, July 2015
9. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, pp. 2181–2187. AAAI Press (2015)
10. Liu, H., Wu, Y., Yang, Y.: Analogical inference for multi-relational embeddings. arXiv: Learning (2017)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, 2–4 May 2013, Workshop Track Proceedings (2013)
12. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. arXiv: Artificial Intelligence (2015)
13. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, NIPS 2013, pp. 926–934 (2013)
14. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1499–1509, September 2015
15. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, 19–24 June 2016, JMLR Workshop and Conference Proceedings, vol. 48, pp. 2071–2080 (2016)
16. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
17. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph and text jointly embedding. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1591–1601. Association for Computational Linguistics, October 2014
18. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI 2016, pp. 2659–2665 (2016)

19. Xu, J., Chen, K., Qiu, X., Huang, X.: Knowledge graph representation with jointly structural and textual encoding. arXiv: Computation and Language (2016)
20. Yosef, M.A., Hoffart, J., Bordino, I., Spaniol, M., Weikum, G.: AIDA: an online tool for accurate disambiguation of named entities in text and tables. Proc. VLDB Endow. **4**(12), 1450–1453 (2011)
21. Zhu, S., Cheng, X., Su, S.: Knowledge-based question answering by tree-to-sequence learning. Neurocomputing **372**, 64–72 (2020)