A Joint Model for Representation Learning of Tibetan Knowledge Graph Based on Encyclopedia

YUAN SUN*, School of Information Engineering, Minzu University of China
ANDONG CHEN, School of Information Engineering, Minzu University of China
CHAOFAN CHEN, School of Information Engineering, Minzu University of China
TIANCI XIA, School of Information Engineering, Minzu University of China
XIAOBING ZHAO, School of Information Engineering, Minzu University of China

Learning the representation of a knowledge graph is critical to the field of natural language processing. There is a lot of research for English knowledge graph representation. However, for the low-resource languages, such as Tibetan, how to represent sparse knowledge graphs is a key problem. In this paper, aiming at scarcity of Tibetan knowledge graphs, we extend the Tibetan knowledge graph by using the triples of the high-resource language knowledge graphs and POI (Point of Information) map information. To improve the represen -tation learning of the Tibetan knowledge graph, we propose a joint model to merge structure and entity description information based on the TransE and CNN model. In addition, to solve the segmentation errors, we use character and word embedding to learn more complex information in Tibetan. Finally, the experimental results show that our model can make a better representation of the Tibetan knowledge graph than the baseline.

CCS Concepts: • **Artificial intelligence** → **Natural language processing**; • **Neural network**;

Additional Key Words and Phrases: Tibetan, Knowledge Graph, Representation Learning, Joint Model, Encyclopedia

ACM Reference Format:
Yuan Sun, Andong Chen, Chaofan Chen, Tianci Xia, and Xiaobing Zhao. 2020. A Joint Model for Representation Learning of Tibetan Knowledge Graph Based on Encyclopedia. *J. ACM* *, *, Article * (* 2020), 17 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Aiming at extracting structured knowledge from unstructured or semi-structured information, knowledge graphs become an important part in natural language processing. A large number of researchers have worked on constructing knowledge graphs, such as Freebase [1], DBpedia [2], YAGO [3]. To form a multi-dimensional graph, they use nodes to represent entities, and edges to

*Corresponding author.

Authors' addresses: Yuan Sun, School of Information Engineering, Minzu University of China, Beijing, 100081, China, tracy.yuan.sun@gmail.com; Andong Chen, School of Information Engineering, Minzu University of China, Beijing, 100081, China, ands691119@gmail.com; Chaofan Chen, School of Information Engineering, Minzu University of China, Beijing, 100081, China; Tianci Xia, School of Information Engineering, Minzu University of China, Beijing, 100081, China; Xiaobing Zhao, School of Information Engineering, Minzu University of China, Beijing, 100081, China, nmzxb_cn@163.com.

0004-5411/2020/*-ART* $15.00
https://doi.org/10.1145/1122445.1122456

represent the relation between entities. But the construction of Tibetan knowledge graph is still in its infancy.

Remarkable success has been achieved in the last few years on representation learning [4] based on deep learning methods [5], such as the TransE, TransR, TransH and other methods [6–14], which translate the triples ($< h, r, t >$) (<head entity, relation, tail entity>) into low-density vectors to represent knowledge graphs. Although these models are effective on English knowledge graphs, for low-resource languages, such as Tibetan, there are few research works.

Currently, there are very few Tibetan knowledge graphs available, and the data is very sparse. In this paper, we construct a Tibetan knowledge graph data set (TD50K) which includes 53,797 triples, 12,573 entities and 3,285 relations. In the knowledge graph, only 48% of entities contain two or more triples. The average number of triples owned by an entity is 2, while in the famous English data set FB15k, 98% of entities have two or more triples, and its average number of triples owned by an entity is 39. Obviously, it can be seen that the Tibetan data is very sparse, which will seriously affect the representation of Tibetan knowledge graphs. Therefore, how to use the rich knowledge graphs resources in other languages to extend the scale of Tibetan knowledge graphs is an important issue. Moreover, despite the construction of Tibetan knowledge graphs, the amount of data is much lower than that of English and Chinese knowledge graphs. So how to represent and learn the knowledge graphs for low-resource languages like Tibetan is another important issue.

To solve these problems, this paper propose a joint model for representation learning of Tibetan knowledge graph based on entity descriptions and structure information. Aiming at the scarcity of Tibetan knowledge graph, we extend the Tibetan knowledge graph by using the triples of high-resource language knowledge graphs and POI map information. Furthermore, we apply the model to the English data, and the effect is also improved.

Our contributions can be summarized as follows:

(1) To expend the scale of Tibetan knowledge graph, according to the coreferential relation, we extend the Tibetan knowledge graph by using the triples of other language knowledge graphs and map POI information. Then a Chinese-Tibetan dictionary is introduced to translate Chinese entity information into Tibetan.

(2) To solve the segmentation errors, we use character and word embedding to learn more complex information in Tibetan.

(3) To better represent the triples in the knowledge graph, we propose a joint model to merge structure and description representation of the knowledge graph. The TransE algorithm is used to generate the structure representation of the Tibetan knowledge graph. An attention mechanism is used to filter important words in descriptions from Tibetan encyclopedia corpus, and the CNN model trains the co-occurrence matrix through descriptions information filtered.

2 RELATED WORK

Recently, deep learning methods have gotten remarkable success in artificial intelligence. These great achievements are inseparable from the representation of data. For the natural language processing community, representation learning has gradually become a hot topic. On the other hand, by extracting the knowledge of human society and storing it in a structured manner, the knowledge graph supports the useful application of artificial intelligence such as question answerin -g, knowledge inference and so on. Therefore, many researchers have begun to focus on the study of knowledge graph construction [15–18] and representation [19, 20].

The main part of knowledge graphs construction is knowledge extraction. Based on the existing knowledge graph, Mihalcea et al. and Milne et al. [21, 22] used the information retrieval method to generate entity links. Bunescu et al. [23] established relations between entities and multiple representations by extracting pages from Wikipedia, and used redirections and information box

to set up knowledge graph. The above researches are based on pattern matching to extract the fixed knowledge. To extract knowledge more flexibly, Chinatsu et al. [24] proposed a dictionary-driven knowledge extraction method. By extracting the entity knowledge within the dictionary, the method extended the English knowledge graphs. Recently, deep learning methods are widely used in knowledge extraction and other natural language processing [25–29]. LeCun et al. [30] proposed the Convolution Neural Networks (CNN), which is an important feature extraction netwo -rk. CNN includes convolution and pooling layers. The convolution layer mainly performs feature extraction, while the pooling layer mainly reduces the number of extracted features. With the two strong components, CNN can accurately extract features. Zeng et al. [31] implemented knowledge extraction through CNN, and Wang et al. [32] introduced an attention mechanism to extract knowledge. But these methods are based on a large number of corpora, and it is difficult to achieve high performance for low-resource languages such as Tibetan.

Based on multi-class mapping, there are many semantic representation methods [7, 33]. These methods mainly treat the triples <head entity, relation, tail entity> as a translation work. In 2013, Bordes et al. proposed the TransE [6] to represent the knowledge graph. They regarded the relations of the knowledge graph as a kind of translation vector between entities. With high computational efficiency, their model got good performance on one-to-one relation. Unfortunately, when dealing with "one-to-N", "N-to-one", and "N-to-N" relations, their model do not perform well. In order to solve this problem, many researchers have proposed a series of improvement methods. The core idea is to set a mapping function according to the type of semantic relation. By mapping the head and tail entities with the semantic relation space, lots of works were proposed to learn the representation of knowledge graphs [7, 11–13]. After that, researchers have proposed the TransH and TransR models [8, 9], which are an improvement on the TransE model. TransH puts the head and tail entities vectors on the hyperplane where the relations are located, and then completes the translation process. Different from TransH, TransR model is to define a matrix for each relation, which is used to transform the entity vector into the relation space, and then to translate it into the relation vector space. TransD model [10] is an improvement on the TransR model. It believes that a fixed-size transformation matrix in TransR should be dynamically determined by the entity-relation pairs. At the same time, the model considers entity and relation as a vector operation rather than matrix operation in TransR. Therefore, TransR greatly reduces the amount of calculation. These improved models got about 50% Hits@10 when predicting multiple entities, while the result is unsatisfactory. TransE, TransH and TransR all believe that each relation can only represent single semantic representation. However, the relation may represent different meanings in fact. For example, the relation in the triples <Kunlun Mountain, Relation, China> can describe mountain or area. Towards that end, researchers are inspired by the distributed perspective of knowledge representation [7, 34], and adopt a method based on the TransE. KG2E [14] is the typical example. It proposes a distribution-based representation learning on the TransE model. KG2E uses Gaussian methods to learn the representation of entities and relations in the knowledge graph. Different from the above models, KG2E supposes the entities, categories, and relations obey Gaussian distribu -tion. By introducing the co-variance matrix, the KG2E can represent the uncertainty of entities and relations in the knowledge graph, especially for the relations of one-to-N and N-to-one.

Some researchers have proposed some methods to fuse the text and knowledge graphs. Wang et al. [8] have introduced a knowledge representation learning model combining knowledge graphs and texts. Their work is based on translation scheme and Skip-gram model, which respectively represent knowledge space and text space. Additionally, the model has designed another alignment module that uses Wikipedia text information and entity information to align knowledge and text space. Based on their work, Zhong et al. [35] introduced the auxiliary description of the entity to align the entity vector with all word vectors as close as possible. Similarly, Zhang et al. [36] also

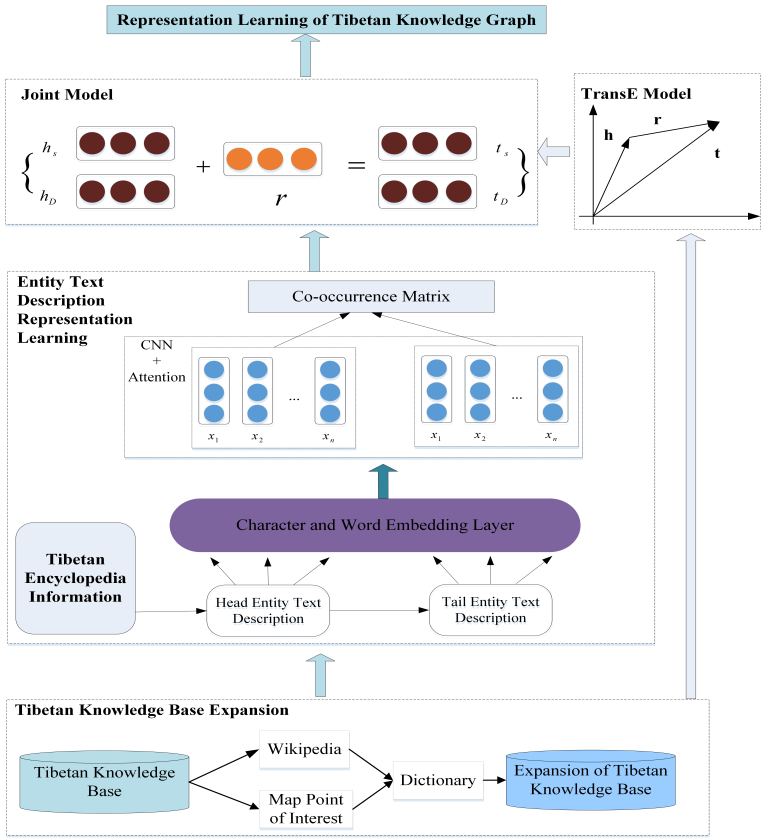

Fig. 1. The Framework of Representation Learning of the Tibetan Knowledge Graph.

tried to represent the entity vector by means of the entity and the entity description vector. The above methods have achieved good performances in knowledge graph representation learning. However, when a new triple is added to the knowledge graph, it needs to be relearned in the global space, which will consume a lot of time.

3 MODEL ARCHITECTURE

In Tibetan knowledge graph representation, we face two problems: (1) Tibetan knowledge graph is very sparse. (2) Traditional algorithms cannot fully represent the entity triples. This paper proposes a joint model, and the framework is shown in Fig.1. For description representation learning, we mainly combine Tibetan encyclopedia to train the co-occurrence matrix of the entities description. For structure representation learning, we use the TransE model to obtain the structure representati -on of the triples. Finally, the two representations are jointly trained to obtain the final representatio -n of the knowledge graph.

This model has four main parts:

(1) Tibetan knowledge graph expansion: We use the triples of high-resource language knowledge graphs and map POI information to extend the knowledge graph, and finally use a Chinese-Tibetan dictionary to translate them into Tibetan.

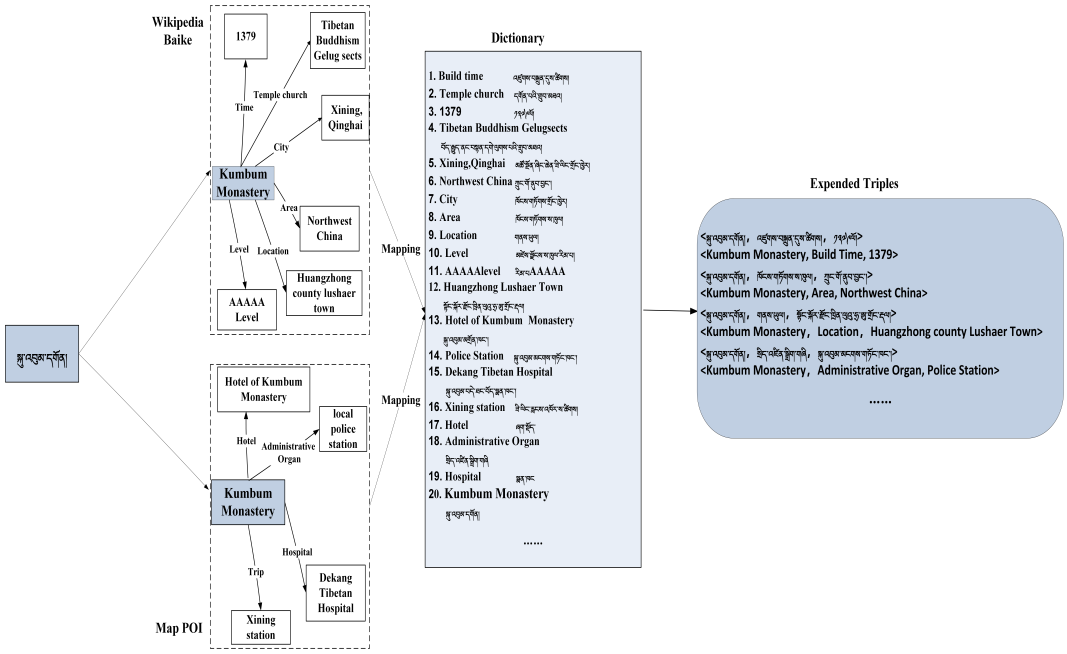

Fig. 2. Expansion of the Tibetan Knowledge Graph.

(2) Entity text description representation: To learn more complex information in Tibetan, we use character and word embedding. In addition, an attention mechanism based on CNN model is used to filter out important words in descriptions from Tibetan encyclopedia corpus and train the co-occurrence matrix.

(3) Structure representation: We uses the TransE model to learn from the knowledge graph triples to obtain the structure entity representation.

(4) Joint model: To get the final knowledge graph representation, we use a joint model to train the description and structure representations.

4 MODEL DETAILS

4.1 Tibetan Knowledge Graph Expansion

For low resource languages, the scale of extracting triples from text corpus is limited. In the knowledge graph built earlier by us, only 48% of entities contain two or more triples. An entity has an average of 2 triples. So how to extend the knowledge graph is very important. In this paper, we use the coreferential relation in Tibetan knowledge graph to get the entity information in Baike and Wikipedia and extend the triples. The extended method is shown in Fig.2. In addition, with the map POI information, we can get other relations of Chinese entities, then transfer them to Tibetan entities to extend the Tibetan knowledge graph.

4.1.1 Using Chinese Encyclopedia Data. For example in Fig.2, the entity ཀླུ་འབུམ་དགོན། (Kumbum Monas -tery) only includes one triple <ཀླུ་འབུམ་དགོན།, རྒྱ་ཡིག་གི་མིང་། (Chinese name)， 塔尔寺 (Kumbu Mnastery)>. We search the triples of 塔尔寺 (Kumbum Monastery) in Baike and Wikipedia and get the related triples. Then we use a Chinese-Tibetan dictionary and the NiuTrans [37] system to translate the Chinese words into Tibetan, and extend Tibetan entity relations. The Chinese-Tibetan dictionary

is edited by Zhang Yisun, published by House of Minority Nationalities in 1993, which includes about 53,000 words.

In the above process, there are two problems we need to solve.

(1) Some of Chinese entities map two or more entities in the Baike and Wikipedia, how to determine which is the correct entity in Chinese corresponding to the Tibetan entity is the first problem. For example, when we search for < 海豚 > (dolphin), there are two entity mappings. One is the animal dolphin, the other is a movie's name dolphin. However, in Tibetan, the triples of dolphin are all about the animal. In order to solve the problem, we translate all the triples of dolphin in Tibetan into Chinese, and use the word2vec tool [38] to calculate the similar distance of dolphin's triples in two meanings. If the similar distance between translated triples and animal dolphin triples from Baike and Wikipedia is closer than the movie name dolphin, we can determine that the animal dolphin is the correct entity corresponding to the Tibetan entity. Moreover, if there is only one triple in Tibetan, we use the description information of the entity, extract the keywords by TF-IDF method and translate them into Chinese. Then using the above method to find the correct entity in Chinese.

(2) Another problem is that we have to do deduplication after we get the triples. Sometimes Baike data and Wikipedia data have the same meaning in the triples, but the words are different. For example, in triples of entity < 海豚 > (dolphin), one relation is < 进化支 > (Evolutionary Branch) and the attribute is < 海豚形类 > (Delphinida) in Baike. But in Wikipedia, the relation is < 演化支 > (Evolutionary Branch), but the attribute is also < 海豚形类 > (Delphinida). So, in this case, we choose one of expressions in the relation since they have the same attribute value.

4.1.2 Using POI Information. POI information can also extend the Tibetan knowledge graph, the process is as follows:

(1) We use Chinese entities in Tibetan triples to retrieve POI information through the POI API interface provided by Baidu Maps.

(2) If a returned data has the 'pois' field, it means that the entity has relevant POI information in Baidu map. Otherwise, it means that the entity does not have geographic information.

(3) From the 'pois' field, we use the regular expression to extract the corresponding triple information. For example, when searching for <བཀྲ་ཤིས་ལྷུག་ཁྱང> (auspicious and fertile ditch), we can retrieve <རང་སྐྱང་མཛེས་སྤྱོངས་ཁུལ> (natural scenic spot) by tag (address type).

There are also several problems when we use POI information to extend the triples of Tibetan. First of all, when using Baidu map, an entity of POI information can be predicted automatically and a mass of relevant information can be shown, so we need to choose the most relevant information. In this paper, we choose the first information provided and use its POI information to extend triples. Secondly, when the map returns some information with the same name, but they are different places, we use triples information about the entity to search POI information of different places. If we can match a same word in a place, we will regard the place as a correct POI information.

4.2 Entity Text Description Representation Learning

The entity text description contains rich entity information, which can be used as an extra informati -on in the knowledge graph that can improve the representation learning effect. For example, in Fig.3, the triples<ཡུས་ཀོ, བརྩམས་རྩོམ, བ་ཡིའི་འཐག་མ་སྤྱིང> (<Hugo, author, Notre Dame de Paris>) and related entity descriptions are given. It is not difficult to find that many entities in the description informati -on imply the information triples. If we can obtain the co-occurrence matrix automatically by CNN and incorporate it into the knowledge representation learning, the performance of the triple knowledge representation will undoubtedly be improved. However, most current knowledge repres -entation learning models only focus on the structured triples in the knowledge graph, and ignore

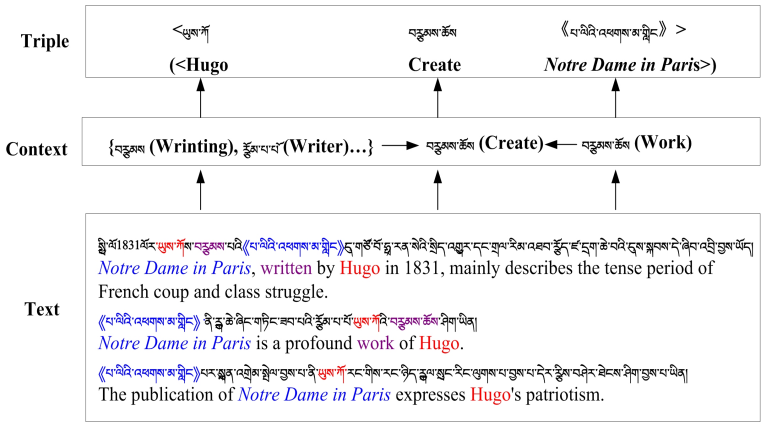

Fig. 3. Example of Entity Text Description Information.

Table 1. Word-level and Character-level Segmentation.

Original Sentence	Character-level	Word-level
སྒོར་མཁན་ཀྱི་རྩོལ་སྐྱེལ་ལ་གཞིགས	སྒོར་ / མཁན / ཀྱི / རྩོལ / སྐྱེལ / ལ / གཞིགས	སྒོར་མཁན་ / ཀྱི་ / རྩོལ་སྐྱེལ་ / ལ་ / གཞིགས
དྲ་ངོས་ཏྲན་འགོད་ནི་ཞེ་ལས	དྲ / ངོས / ཏྲས / འགོད / ནི / ཞེ / ལས	དྲ་ངོས་ / ཏྲས་འགོད་ / ནི་ / ཞེ་ལས

the potential information of entity description for representation learning. Therefore, it is necessary to make full use of the entity described information.

In order to obtain more complex information in Tibetan, we use character and word embedding. As we all know, the information carried by the text is rich; however, not all the words in the text are related to the corresponding entities in triples. Therefore, when introducing text description, it is inevitable to introduce noise that is irrelevant word to the corresponding entities, so how to remove the noise from the text description is a problem that must be considered. In this paper, we introduce the attention mechanism, which will give more weight to the words related to the corresponding entities.

For co-occurrence matrix, we use two-layer CNN to generate text description representation of entities. Then we fuse the description representations and character and word embedding of entities to generate the co-occurrence matrix.

4.2.1 Character and Word Embedding Layer. Tibetan is a pinyin language, so character is the smallest unit of word. However, there are some segmentation errors in Tibetan, so we use character and word embedding to learn more complex information in Tibetan. For character-level, we can use '·' to segment the corpus into character-level data set. For word-level, we use the Tibetan word segmentation tool [39]. The results are shown in Table 1.

We use the word2vec model [38] to generate embedding vector in word-level corpus. As for character-level corpus, we use a bi-direction long short-term memory neural network (BiLSTM) and obtain character-level embedding.

Finally, we fuse character-level and word-level embedding through a high-way network which has two layers. We use a lookup method to get embedding about entities in the triples and words in the text description.

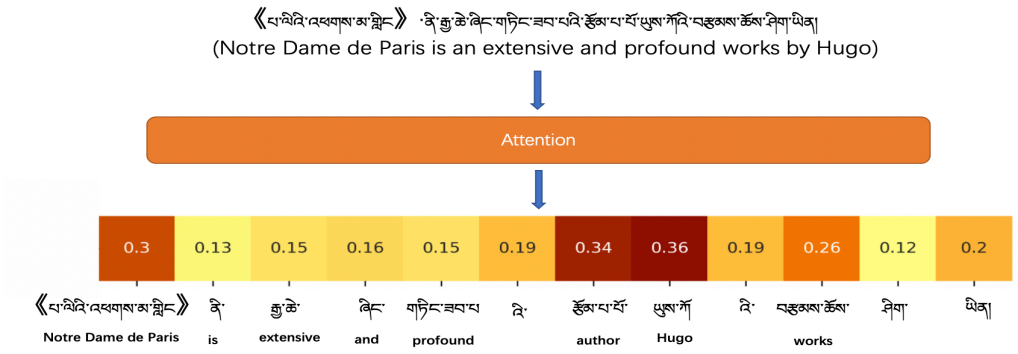

Fig. 4. Example of Attentions in Description Information.

4.2.2 Attention Mechanism. In the preproccessing stage, in order to eliminate the noise of irreleva -nt words and obtain important words in the text, we adopt an attention mechanism, which can assign low scores to irrelevant words and high scores to relevant words. An example is shown in Fig.4. In the description of the entity ཡུས་ཀོ (Hugo), the weights of བ་ལིའི་འཕ (Notre Dame de Paris) and འཕ་རྩམས་རྩོམ (works) are higher than other words. The weight score is calculated by the formula (1).

$$\xi(\mathrm{D(c)}, \mathrm{T(e)}) = \mathrm{W}_{\mathrm{D(c)}} \circ F_w\left(\mathrm{W}_{\mathrm{D(c)}}, \mathrm{W}_{\mathrm{T(e)}}\right) \tag{1}$$

where e is an entity, c is a word from a text description. T and D represent mapping function about text description and vocabulary of entities. $\mathrm{W}_{\mathrm{D(c)}} \in \mathbb{R}^{|\mathrm{D(c)}|\times k}$ represents text description matrix about words, where each line represents k dimensional embedding for a word. $\mathrm{W}_{\mathrm{T(e)}} \in \mathbb{R}^{|\mathrm{T(e)}|\times k}$ entity matrix is about entities, where each line represents k dimensional embedding for an entity. F_w is a function which calculates weight score for each row. To obtain the weight score, we define F_w in the formula (2).

$$F_w\left(\mathrm{W}_{\mathrm{D(c)}}, \mathrm{W}_{\mathrm{T(e)}}\right)_{[i]} = \max_j \left(\frac{\sum_m^k \mathrm{W}_{\mathrm{D(c)}[i,m]} \mathrm{W}_{\mathrm{T(e)}[j,m]}}{\sqrt{\sum_m^k \mathrm{W}^2_{\mathrm{D(c)}[i,m]}} \sqrt{\sum_m^k \mathrm{W}^2_{\mathrm{T(e)}[j,m]}}} \right) \tag{2}$$

The i^{th} word weight score in $\mathrm{D(c)}$ is the largest cosine similarity score between the embedding of the i^{th} word in $\mathrm{W}_{\mathrm{D(c)}}$ and the embedding of the corresponding entity in $\mathrm{W}_{\phi(e)}$.

This function gives lower scores to words that are irrelevant to the corresponding entity, and higher scores to words that are relevant and important. $\circ$ is row-wise product, so $\xi(\mathrm{D(c)}, \mathrm{T(e)})$ represents the importance of all words in text descriptions.

Finally, the text description information only includes the relevant and important words.

4.2.3 Convolutional Neural Network. It is the core operation of the CNN. We use the one dimensio -nal convolution kernel of length l. The convolution layer has two main operations. The convolution layer performs window operations. X^l is set to be the input of l^{th} convolution layer. To be specific, $X^{(1)}$ is set of vectors $\left\{x_1^{(1)}, x_2^{(1)}, \cdots, x_n^{(1)}\right\}$ which is the input in the first layer. We have the following window operations, as is shown in the formula (3).

$$x_i'^{(l)} = x_{i:t+k-1}^{(l)} = \left[x_i^{(l)^T}, x_{i+1}^{(l)^T}, \cdots, x_{i+k-1}^{(l)^T}\right]^T \tag{3}$$

$x_i'^{(l)}$ is the i^{th} vector of the matrix after the window operation, and k is the window size, which determines the size of the convolution kernel of this layer. It can be seen that the window operation connects the i vector and the $i+k-1$ vector to form a long vector. Considering the input sequence may be variable in length, we use the method of zero padding strategy.

After the window operation, the convolution layer combines the convolution kernel to get the output $Z^{(l)} = \left\{z_1^{(l)}, z_2^{(l)}, \cdots, z_n^{(l)}\right\}$. The formula is shown in (4).

$$z_i^{(l)} = \sigma\left(W^{(l)} x_i'^{(l)} + b_i^{(l)}\right) \tag{4}$$

where $z_i^{(l)}$ represents the i^{th} output vector of the l layer, $W^{(l)}$ is the convolution kernel of the l layer, and $b_i^{(l)}$ represents the bias matrix of the l layer. σ is a non-linear function, such as ReLU and tanh. It should be noted that the zero-padded strategy has no effect on the result during forward and backward propagation.

After the convolution operation, the pooling layer is needed to reduce and filter the unnecessary features. Differently, this paper uses different pooling strategies. In the first pooling layer, we use the maximum pooling strategy. We get the output matrix $Z^{(l)}$ by passing the convolution and the non-linear layer. We divide this matrix into vector groups of same size according to the rows, and every size of vector is $m \times n_2$. For each dimension, we choose the vector with largest size among the m elements to form the output vector with size of $1 \times n_2$. Formally, we have the formula shown in (5).

$$x_i^{(2)} = \max\left(z_{n \cdot i}^{(1)}, \cdots, z_{n \cdot (i+1)-1}^{(1)}\right) \tag{5}$$

In the above formula, *max* represents the operation taking the maximum value. Through the maximum pooling strategy, we select the strongest feature signal in the local area.

In the second pooling layer, we use the average pooling strategy instead of the maximum pooling strategy. Since we need to get a description-based representation of the entity, this pool layer should output a vector. Specifically, we have the formula shown in (6).

$$x^{(3)} = \frac{1}{n_2} \sum_{i=1}^{n_2} z_i^{(2)} \tag{6}$$

$x^{(3)}$ is the entity description representation from the CNN, and n_2 represents the input matrix length of the second pooling layer. The above formula shows that the average pooling layer takes the input matrix $z^{(2)}$ as the average of the rows to form the output vector.

As is mentioned above, we set different pooling strategies on the two pooling layers for the specific task of text description encoding. The maximum pooling is performed in the first pooling layer. Therefore, only the strongest value of the local feature is used as the representation of the entire localization operation. In the second pooling layer, the average pooling is used to cover all local information.

For co-occurrence matrix, we fuse $X^{(3)=}\left\{x_1^{(3)}, x_2^{3)}, \cdots, x_n^{(3)}\right\}$ and embedding vectors of entities. The fused matrix is regard as co-occurrence, shown in the formula (7).

$$\mathrm{W_{co}} = \mathrm{W_{T(e)}} \circ X^{(3)} \tag{7}$$

4.3 Structure Representation Learning

For the structure representation learning of the Tibetan knowledge graph, this paper uses the TransE algorithm. Given a triple $fact = (h, r, t)$, the TransE model represents the relation as a

translation vector $\vec{r}$ and links the entity vectors $\vec{h}$ and $\vec{t}$ with lower errors, shown in the formula (8).

$$\vec{h} + \vec{r} = \vec{t} \tag{8}$$

Scoring function is defined as the distance between $\vec{h} + \vec{r}$ and $\vec{t}$, shown in the formula (9).

$$f_r(h,t) = -\|\vec{h} + \vec{r} - \vec{t}\|_{\frac{1}{2}} \tag{9}$$

If the triple (h,r,t) exists, the value of the function f_r should be higher.

4.4 Joint Model

The joint model still follows the assumption of the translation model. Based on that assumption, the entities and relations in triples should be in a certain relation. Specifically, the joint model defines the following energy functions, shown in the formula (10).

$$E(h,r,t) = a_1 \|h_S + r - t_S\| + a_2 \|h_S + r - t_D\| + a_3 \|h_D + r - t_S\| \, h_D + r - t_D\| \tag{10}$$

In formula (10), a_1, a_2, a_3, a_4 are hyper parameters that control various weights. In the energy function, the $\|h_S + r - t_S\|$ part is similar to the energy function defined in the translation model, and the three terms $\|h_S + r - t_D\|$, $\|h_D + r - t_S\|$, $\|h_D + r - t_D\|$ are based on the description of the entity vector. Through the soft limit of such mixed terms, the joint model can naturally map the two vectors of the entity to the same semantic space, and can share the same relation vector.

Inspired by the learning framework of the translation model, we use an improved energy functio -n. The maximum interval method is used, and a scoring function is defined to optimize the model, as is shown in the formula (11).

$$L = \sum_{(h,r,t)\in T} \sum_{(h',r',t')\in T} \max\left(\gamma + E(h,r,t) - E\left(h',r',t'\right)\right) \tag{11}$$

where (h',r',t') is a negative triple, and γ is a hyper parameter representing the interval distance of the positive and negative triples. Different from the translation model, the energy function $E(h,r,t)$ of the triple score includes four combination terms represented by two entity vectors. The maximum interval method is to determine that the energy function score of the positive triple is smaller than the negative triple. T' is a set of negative triples. For a given positive triple (h,r,t), we set the negative triple. The set is shown in the formula (12).

$$T' = \{(h',r,t)\,|h' \in E\} \cup \{(h,r,t')\,|t' \in E\} \cup \{(h,r',t)\,|r' \in R\}, \quad (h,r,t) \in T \tag{12}$$

Formula (12) indicates that all negative triples are generated by randomly replacing any entity (or relation) in the positive triple with another entity (or relation). At the same time, in order to reduce the random selection of negative triples, the following restrictions are added when generating negative triples, as is shown in the formula (13).

$$\forall\,(h',r',t') \in T', \quad (h',r',t') \notin T \tag{13}$$

This method not only avoids to generate positive triples, but also greatly reduces the randomness of negative triples, which makes the training model have stronger generalization ability.

Table 2. Data Set Information.

Data Set	Entities	Relations	Triples	Average Number
TD50K	12,573	3,285	53,797	2
TD80K	12,584	4,256	80,178	6
FB15K	14,951	1,345	592,213	39

5 EXPERIMENTS AND RESULTS ANALYSIS

5.1 Data Sets

We conduct some experiments on Tibetan data sets TD50K and TD80K, and the English data set FB15K. Table 2 shows the data in detail.

TD50K: This is the Tibetan data before expansion, which includes 53,797 triples, 12,573 entities and 3,285 relations. This data is built by us. And every entity contains 2 triples on average. The maximum number of triples is 416, and the minimum number is 1. A part of the data can be obtained from *https://github.com/andongBlue/TiData50K/tree/master.*

TD80K: This is the Tibetan data after expansion based on the methods in section 4.1. This data set includes 80,178 triples, 12,584 entities and 4,256 relations. Every entity contains 6 triples on average, and the maximum number of triples in one entity is 508.

FB15K: We also conduct the experiment on the English data set FB15K, which includes 592,213 triples, 14,951 entities and 1,345 relations. Every entity contains 39 triples on average, and the maximum number of triples in one entity is 9,645. The data is available from *https://everest.hds.utc.f -r/doku.php?id=en:transe.*

TD12K: This is the Tibetan data added the description from Yunzang Encyclopedia website. We remove the entities that do not obtain description information from the website. In order to ensure that the description of each entity has enough information, we delete the entity which description information has less than 2 words. This data includes 50,000 triples and 12,449 description informati -on. For the Tibetan encyclopedia knowledge data, the specific format is shown in Table 3, and multiple entities are separated by the "##" symbol. This data set is used to evaluate our joint model.

FB20K [40]: This is the English data including description information for 14,904 entities based on FB15K.

5.2 Evaluation

Following the paper [6], we use two evaluation methods.

(1) MeanRank: MeanRank of correct entities.

For each test set, we remove the head entity and let other entities replace it. We call these replaced triples corrupted triples. Then we put these corrupted triples into the model to make predictions. Next, we sort the predicted entities in ascending order. MeanRank is the average score of the correct predicted triples. The smaller value of MeanRank is, the better the model is.

(2) Hits@10: Proportion of valid entities ranked in top 10.

In some cases, the replaced entity may be the correct triple that appears in the training set and the validation set. In this case, the experimental results may be affected, so we remove those triples that have appeared in the training set and the validation set after processing the test set. The original test set is called "raw" and the processed test set is called "filtered".

Table 5. Experimental Results of Different Models.

	TD50K				TD80K				FB15K			
	MeanRank		Hits@10		MeanRank		Hits@10		MeanRank		Hits@10	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	497	491	24%	25%	456	447	61%	64%	219	178	50%	66%
TransH	490	485	27%	29%	449	449	**64%**	**65%**	212	104	53%	67%
DistMult	**480**	**477**	**30%**	**31%**	**415**	**409**	59%	61%	**210**	**104**	**57%**	**69%**

Table 6. Joint Model Parameter Setting.

Parameter	Value
Learning rate	0.001
Vector dim	100
lambda	1.5
Mini-batch	100

comparison experiment further proves that the sparsity of the triples can affect the representation of the knowledge graph. The expansion method of the triples proposed in this paper can effectively reduce the sparsity of the low-resource knowledge graph, and optimize the low-resource knowledg -e graph representation.

5.4 Experiments on the Joint Model

In the experiment of the joint model based on entity description, we use the entity description data set TD12K. The structured knowledge graph is represented by TransE model, and an attention mechanism is used to filter out important words in descriptions from Tibetan encyclopedia corpus. Then, CNN model trains the co-occurrence matrix through descriptions information filtered. The result of the final knowledge graph representation is the combination of the weights of the two representations.

For data sets, we divide the TD12K into three groups: TD12K-1, TD12K-3 and TD12K-5, which prove the result of our model in different number of data sets. Meanwhile, we experiment on the English data set FB20K based on our joint model. We use the TransE model as the baseline.

TD12K-1: Select 10,000 triples and description information from TD12K randomly.

TD12K-3: Select 30,000 triples and description information from TD12K randomly.

TD12K-5: Use all the 50,000 triples and description information in TD12K.

We compare the proposed joint model with the DKRL [40] model and ConMask [42] model (the baselines), which learn knowledge representations with structure information and description information.

5.4.1 Parameter Setting. We explore the effects of different parameters on the experimental results. In order to obtain better results, we select most suitable parameters. Parts of parameter settings are shown in Table 6. The edge distribution lambda is initially set between $\{0.4, 1, 1.5, 4\}$. We find that if the lambda is set too large, the model is not easy to converge, and the model training takes too long when it is set too small. We set the parameters of the energy function as follows: $a_1 = 1.0$, $a_2 = 0.3$, $a_3 = 0.4$, $a_4 = 0.3$.

Table 3. Data of Tibetan Encyclopedia.

Name	Entities Information	Description Text
སྲོག་ཆགས་སོག་ཆགས།	སོག་ཆགས་ཁམས། ## སོར་ཆོག་སོག་ཆགས་སོ## རེཁས1700	སྲོར་ཆོག་སོག་ཆགས་ནི་གཞོག ས་གཤིས་ཆ་འཕྲིག་ཡིན་པ་དང་ལུས་ཤུང་ ལ་ཆོག་ཀྱི་དཔེ་བ་ཡོང་པའི་སོག་ཆགས་ལ་རེར།
Vertebrate	Annelida ## Vertebrata## 17,000 kinds	Refers to animals with vertebrae, a sub phylum of chordate.
ཏའེ་ཀའོ་ལེ།	ཐུ་རན་ས།## སྨྱི་ལ1890—1970## ཙུང་ཐུང་།##དམག་སྨྱི##ཚབ་སྦིད་པ	སྐབས18འའི་ཐུ་རན་ནེའི་ཙུང་ཐུང་དང་ཏའེ་ཀ འོ་ལེ་རིང་ལུགས། རང་དབང་གི་ཐུ་རན་ས་བཙུགས་ཤིང་ཐུ དམག་གི་དབུ་ཁྲིད་ནས་པ་ཡི་བཅིངས་འཛོལ་བཏོང་ བ། ཐུ་སི་ལན་སྨྱི་མཐུན་རྒྱལ་ཁབ་ཐུ་བ་བཙུགས་པ།
Charles de Gaulle	Charles de Gaulle ## France## AD 1890-1970## president##general##politician	The eighth French president and Charles de Gaulle lead the French liberation army to liberate Paris and establish the fifth Fascist Republic.

Table 4. Structured Model Parameter Setting.

Parameter	Value
Batch	100
Alpha	0.001
Margin	1
Dimension	100
Work-threads	8

5.3 Experiments on Knowledge Graph Expansion

We use TransE [6], TransH [8] and DistMult [41] models to represent the knowledge graph respecti -vely. Tibetan experiments are based on the TD50K and TD80K data sets, and English experiments are based on the FB15K data set. In the experiments, the training set, test set and validation set are divided into 60%, 20% and 20%. And parts of parameter settings are shown in Table 4.

Table 5 shows the experimental results of the knowledge graph expansion. On the TD50K data set, DistMult model gets the best results. The filter of MeanRank value is 477 and Hits@10 value is 31%. On the TD80K data set, the DistMult model also gets the best results in the MeanRank value which is 415, while the TransH model gets the best results in Hits@10 which is 65%.

Having compared the experimental results of the TD50K and TD80K, we find that the values of Hits@10 in three models are higher than 60%, which have increased by 39%, 36% and 30% respectively. In the DistMult model, the MeanRank value drops to 409, which shows that the expansion of Tibetan triples can effectively improve the representation of Tibetan knowledge graph.

Meanwhile, we also carry out experiments on the English corpus FB15K, shown in Table 5. And all values of MeanRank are below 220, and the best value of Hits@10 is 69%. It can be clearly seen that the metrics of the high-resource data are better than the low-resource data. This

Table 7. Experimental Results of Three Model on Tibetan and English.

Model and Data			MeanRank Raw	MeanRank Filter	Hits@10 Raw	Hits@10 Filter
Tibetan	TransE	TD80K	456	447	61%	64%
	DKRL	TD12K-1	452	445	69%	71%
		TD12K-3	446	442	72%	76%
		TD12K-5	426	397	76%	78%
	ConMask	TD12K-1	431	421	62%	64%
		TD12K-3	412	410	77%	78%
		TD12K-5	381	377	78%	79%
	Joint Model	TD12K-1	438	432	68%	70%
		TD12K-3	410	408	78%	80%
		TD12K-5	**376**	**371**	**79%**	**80%**
English	TransE	FB15K	219	178	50%	66%
	ConMask	FB20K	**116**	107	62%	**69%**
	DKRL	FB20K	116	104	52%	67%
	Joint Model	FB20K	118	**106**	**63%**	69%

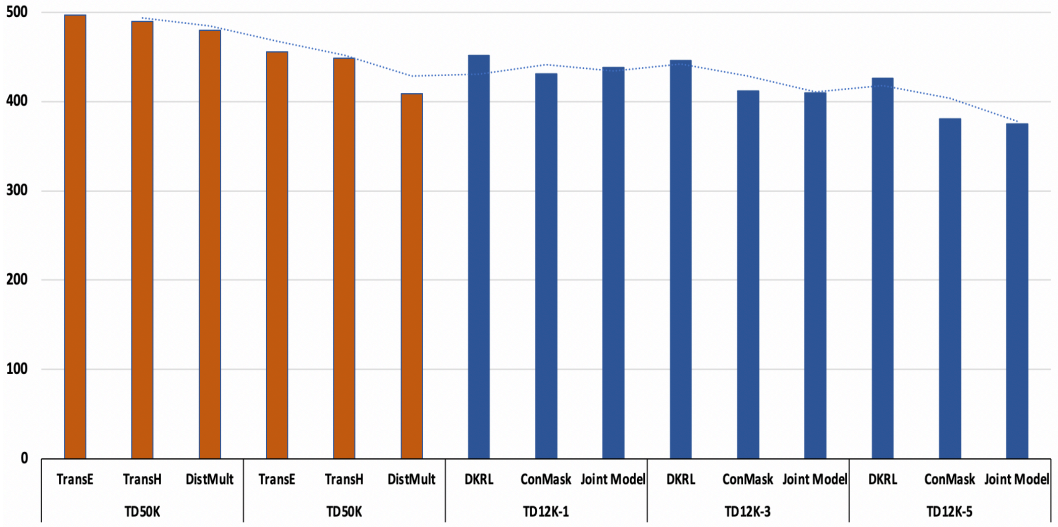

Fig. 5. Comparison of Different Models in MeanRank.

5.4.2 Experimental Results and Analysis. Table 7 shows the experimental results. We can observe from the experiment.

On the TD80K, MeanRank and Hits@10 in the TransE are 456 and 61% respectively, while the DKRL, ConMask and joint model are higher than the TransE model. On the TD12K-1, the ConMask has the highest MeanRank which is 412, while the highest score of Hits@10 is the DKRL. As the number of data increases, our model keeps the best results. On the TD12K-3, MeanRank and Hits@10 in our model are 410 and 78% respectively, which are better than the DKRL and ConMask. On the TD12K-5, the result of MeanRank reduces by 34 and the result of Hits@10 increases by 1% in our model compared with in the TD12K-3, which is the best results in the Tibetan corpus.

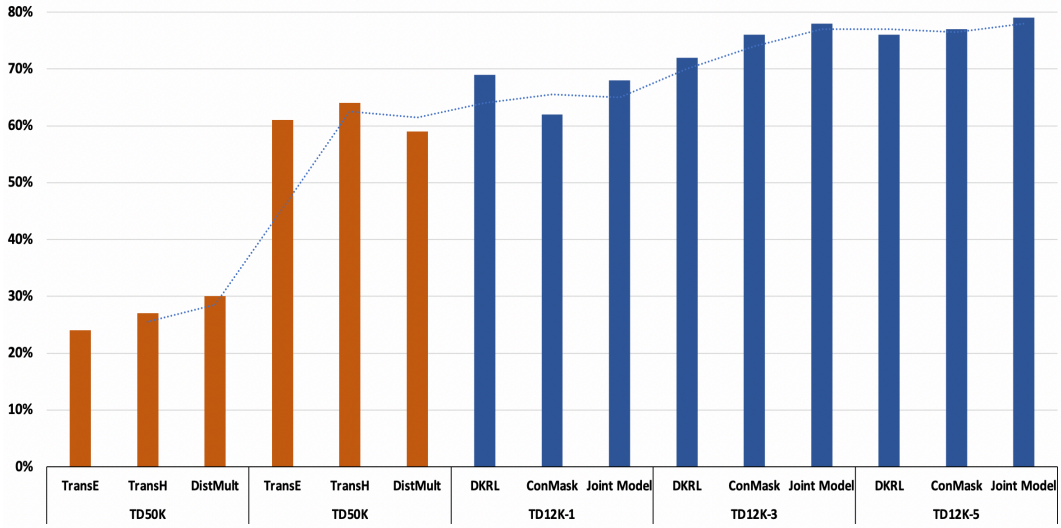

Fig. 6. Comparison of Different Models in Hits@10.

Meanwhile, we also experiment on the English data set FB15K and FB20K. FB20K is an English data set which introduces text description information. The values of MeanRank and Hits@10 using TransE are 219 and 50% respectively. For the ConMask model, the MeanRank is 116, which is the best result. However, our model performs better on Hits@10, which value is 63%.

For the experimental result, we find that our model have better performance, among the other four models. It can be analyzed from the perspective of model architecture that the DKRL model introduces much noise, because it does not use an attention mechanism, and the ConMask does not use character and word embedding, which may loss many semantic information.

Fig.5 shows the comparison of different models in MeanRank on Tibetan. Fig.6 shows the compar -ison between different models in Hits@10 on Tibetan. These results show that the experimental results have greatly improved after extending the triples. The best results appear in the joint model, which indicates that the joint model can improve the representation of the Tibetan knowledge graph.

6 CONCLUSION AND FUTURE WORK

This paper focuses on the expansion and representation learning of Tibetan knowledge graphs. We propose a joint model to integrate structure representation with entity text description representat -ion. The experimental results show the effectiveness of our model. In the future, we will extend the Tibetan knowledge graph and focus on the representation learning when new triples added.

7 ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grant 61972436.

REFERENCES

[1] 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.

[2] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted

from wikipedia. *Semantic Web*, 6(2):167–195, 2015.
[3] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706. ACM, 2007.
[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.
[5] Y Bengio. Learning deep architectures for ai: Foundations and trends® in machine learning, 2, 1–127, 2009.
[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.
[7] Zhiyuan Liu, Maosong Sun, Yankai lin, and Ruobing Xie. Research on knowledge representation learning (in chinese). *Journal of Computer Research and Development*, 53(2):247–261, 2016.
[8] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
[9] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI Conference on Artificial Intelligence*, 2015.
[10] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 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)*, pages 687–696, 2015.
[11] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[12] Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. Transa: An adaptive approach for knowledge graph embedding. *arXiv preprint arXiv:1509.05490*, 2015.
[13] Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. Transg: A generative mixture model for knowledge graph embedding. *arXiv preprint arXiv:1509.05488*, 2015.
[14] He Shizhu, Liu Kang, Ji Guoliang, Zhao Jun, et al. Learning to represent knowledge graphs with gaussian embedding. 2015.
[15] Xiaobin Zhao, Lirong Qiu, and Tiejun Zhao. Multi-nation language ontology knowledge base construction technology (in chinese). *Journal of Chinese Information Processing*, 25(4):71–75, 2011.
[16] Zhen Zhu and Yuan Sun. Tibetan character attribute extraction based on svm and generalization template collaboration (in chinese). *Journal of Chinese Information Processing*, 29(6):220–227, 2015.
[17] Tianci Xia and Yuan Sun. Research on tibetan entity relationship extraction method based on joint model (in chinese). *Journal of Chinese Information Processing*, 32(12):76–83, 2018.
[18] Lili Guo and Yuan Sun. Tibetan person attributes extraction based on bp neural network. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 132–142. Springer, 2016.
[19] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *Artificial Intelligence and Statistics*, pages 127–135, 2012.
[20] 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.
[21] Andras Csomai and Rada Mihalcea. Linking documents to encyclopedic knowledge. *IEEE Intelligent Systems*, 23(5):34–41, 2008.
[22] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 509–518. ACM, 2008.
[23] Razvan Bunescu and Marius Paşca. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.
[24] Chinatsu Aone and Mila Ramos-Santacruz. Rees: a large-scale relation and event extraction system. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 76–83. Association for Computational Linguistics, 2000.
[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
[26] Yuan Sun, Like Wang, Chaofan Chen, Tianci Xia, and Xiaobing Zhao. Improved distant supervised model in tibetan relation extraction using elmo and attention. *IEEE Access*, 7:173054–173062, 2019.
[27] ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. Convolution neural network for relation extraction. In *International Conference on Advanced Data Mining and Applications*, pages 231–242. Springer, 2013.
[28] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation classification via convolutional deep neural network. 2014.
[29] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, 2015.

[30] Yann LeCun et al. Generalization and network design strategies. In *Connectionism in perspective*, volume 19. Citeseer, 1989.
[31] Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*, 2015.
[32] Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 1298–1307, 2016.
[33] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[34] Mengfei Shi. Research and implementation of question answering system based on chinese knowledge base (in chinese). Master's thesis, East China Normal University, 2018.
[35] Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 267–272, 2015.
[36] Dongxu Zhang, Bin Yuan, Dong Wang, and Rong Liu. Joint semantic relevance learning with text data and graph knowledge. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 32–40, 2015.
[37] Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. Niutrans: an open source toolkit for phrase-based and syntax-based machine translation. In *Proceedings of the ACL 2012 System Demonstrations*, pages 19–24, 2012.
[38] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
[39] CJ Long, HD Liu, MH Nuo, and J Wu. Tibetan pos tagging based on syllable tagging. *J. Chin. Inf. Process.*, 29(5):211–216, 2015.
[40] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[41] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.
[42] Dat Quoc Nguyen. An overview of embedding models of entities and relationships for knowledge base completion. *arXiv preprint arXiv:1703.08098*, 2017.