Caps-OWKG: a capsule network model for open-world knowledge graph

Yuhan Wang¹ · Weidong Xiao¹ · Zhen Tan¹ · Xiang Zhao¹

Received: 20 April 2020 / Accepted: 9 December 2020 / Published online: 4 January 2021
© The Author(s) 2021

Abstract

Knowledge graphs are typical multi-relational structures, which is consisted of many entities and relations. Nonetheless, existing knowledge graphs are still sparse and far from being complete. To refine the knowledge graphs, representation learning is utilized to embed entities and relations into low-dimensional spaces. Many existing knowledge graphs embedding models focus on learning latent features in close-world assumption but omit the changeable of each knowledge graph. In this paper, we propose a knowledge graph representation learning model, called Caps-OWKG, which leverages the capsule network to capture the both known and unknown triplets features in open-world knowledge graph. It combines the descriptive text and knowledge graph to get descriptive embedding and structural embedding, simultaneously. Then, the both above embeddings are used to calculate the probability of triplet authenticity. We verify the performance of Caps-OWKG on link prediction task with two common datasets FB15k-237-OWE and DBPedia50k. The experimental results are better than other baselines, and achieve the state-of-the-art performance.

Keywords Capsule network · Open-world knowledge graph · Link prediction · Knowledge graph completion

1 Introduction

Since Google first introduced the concept of knowledge graph (KG) in 2012, it has speed developed in the field of artificial intelligence at an extremely fast speed. Nowadays, research and work related to knowledge graphs have played a pivotal role in various fields under the direction of artificial intelligence. For itself, knowledge graphs are usually stored in triplets (*head, relation, tail*) which show the structure and relationships inside the graph. In recent years, due to the uncompleted and sparse, the researches on knowledge graph about representation learning (KGRL), relationship prediction, and graph completion have attractive many scholars. The proposed models, such as TransE [1], DistMult [23] and ComplEx [18] has achieve state-of-the-arts performance. Besides, some basic models leverage the capsule network [7] to judge the authenticity of the triplets, which can filter the negative triplets and improve the precision of KGs.

But, above models omit the KG structures in the reality world, where the knowledge graphs are dynamic and changeable, and often need to link new entities to complete and update the knowledge graphs. These entities are unknown for existing knowledge graphs, but we also need to judge whether they have a relationship with other entities in the graph.

Hence, the traditional representation learning model in close-world assumption only leverage existing entities to complete the knowledge graph, which is hard to predict the relation between known and unknown entities.

For instance, there is a question “Is Beijing the capital of china?” To answer this question, we need to judge whether the triplets (Beijing, capital_of, China) are true. If Beijing in the triplet is not an entity in the KG, then in this case, this KG is defined as an open-world knowledge graph. Now, in this case, we will utilize the descriptive text of the entity to help determine the authenticity of the triplets.

Recently, several scholars try to represent the knowledge graph based on open-world assumption, which combines the descriptive text of the entity to capture the unknown entity features, such as ConMask [15] and OWE [13]. ConMask need implicit reasoning based on the context, so the evaluation of link prediction depends on the length of descriptive

✉ Xiang Zhao
xiangzhao@nudt.edu.cn

¹ Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, Hunan, China

text greatly. The lack of text will cause a sharp drop in accuracy. In model OWE, it use the embeddings of descriptive text to instead the embeddings of the unseen entities directly. It is not well to fuse the structural embedding and descriptive embedding of triplets.

In this paper, we present a model based on a capsule network to solve the problem of determining triplets of open-world knowledge graphs called Caps-OWKG. For a triplet (head, relation, tail), we obtain the structural embeddings of entities and relationships through the representation learning model of the knowledge graph, and the descriptive texts of the entities are leveraged to obtain the descriptive embeddings of the entities. Then we utilize the pre-trained fusion model to combine the structural and descriptive embeddings to generate get a new triplet embedding. In the end, we use a pre-trained capsule network to process the new triplet embedding representation to obtain the probability of triplet authenticity. With the judgment result of the triplets, we can use this result to complete the tasks of graph completion and relationship prediction.

In this paper, our contributions include:

- In our model, we propose capsule network-based model which leverage the context to represent the relations and entities in open-world knowledge graph [13].
- The basic unit of our model is triplets. The text processing and fusion processing are performed on the triplets and the capsule network also performs feature extraction and judgment on triplets. The advantage of our method is that the triplets can ensure better association between entities and relationships in the triplets during processing. The processed embedding representations of the same entities in different triplets are also fine-tuned, so our model can be regarded as dynamic embedding presentation and can better represent the triplets.
- Our model Caps-OWKG performs experiments on two common datasets FB15k-237-OWE [13] and DBPedia50k [13] with other algorithms such as ConMask [15], TransE-OWE [13] and DKRL [21], and achieves state-of-the-arts results.

The organization of this paper is shown as follow. In Sect. 2, we briefly introduce the related work and then introduce our proposed model Caps-OWKG in detail in Sect. 3. Afterwards, Sect. 4 shows the analysis of our model components and the results of experiments. In the end, we conclude our paper and make a suggestion for future work in Sect. 5.

2 Related work

2.1 Knowledge graph representation learning model

With the expansion of the scale of knowledge graphs, to adjust to large-scale knowledge graph, people have successively proposed a variety of representation learning algorithms based on structural features (triplets) of knowledge graphs. These algorithms embed the entities and relationships into a low-dimensional dense space, and then calculate the associations between the entities and relationships in the space.

The most classic method is the translation-based model TransE [1] proposed by Bordes et al in 2013. The model considers the correct triplet (head, relation, tail) must satisfy $h + r \approx t$ (h represents the embedding of head entity, r represents the embedding of relation and t represents the embedding of tail entity), that is, the tail is obtained by the head through relation translation. TransE is not only simple and efficient, but also has good scalability. In order to overcome the shortcomings of TransE handling 1-to-N, N-to-1 and N-to-N relationship, TransE has been expanded by other models such as TransH [20], TransR [4] and so on. TransH solves this problem by introducing relation-specific hyperplanes and TransR shares the same ideas with TransH. The difference of them is that TransH introduces relation-specific space which is not hyperplanes.

In addition to the simple translation model, RESCAL [8] proposes the bilinear model and the score function of a triplet (head, relation, tail) in it is shown as follow

$$f_r(\text{head}, \text{tail}) = h^T M_r t \quad (1)$$

in which h , t are embeddings of the entities and M_r denotes as a matrix associated with the relation. Yang [23] restricts M_r to diagonal matrices and gets new model DistMult [23]. Then Nickel proposes HolE [9] which can not only have the expression ability of RESCAL but also the efficiency and simplicity of DistMult. And Trouillon proposes ComplEx [18] which introduces complex-valued embeddings to expand DistMult. It can model asymmetric relations better.

Different from the topology-based model which has been widely studied, SME [2] uses neural network architectures to obtain semantic matching. For a triplet (head, relation, tail), it first inputs the embeddings of entities and relations in the input layer. Then in the hidden layer it combines the relation r with the head h to obtain $g_u(h, r)$ and with the tail t to obtain $g_v(t, r)$. Finally, it calculates the dot product of g_u and g_v to get the score of the triplet. Socher proposes a neural network architecture model named NTN [16] which is different from SME. But it is not efficient to adapt to large-scale KGs.

In this paper, it is convenient for us to compare with other open-world KG completion models, so we choose TransE to process the knowledge graph.

2.2 Word embedding model

Word embedding technology converts words in natural language into dense vectors. Similar words will have similar embedding representations. It is convenient for these conversion to mine features between words and sentences in text. There are some classic word embedding models such as Word2vec [5], Glove [10], ELMo [11] and BERT [3] in recent years.

Word2vec is proposed by Mikolov [5] and its core idea is to obtain embedding representation of a word by the context. Two methods are used in the model. CBOW [5] predicts the center word through its nearby words and Skip-gram [5] predicts the nearby words through their center word. Word2vec uses two different optimization strategies hierarchical Softmax and negative sampling.

Word2vec only considers the local information of the word but ignores the global information. Pennington [10] proposes a new model Glove to deal with this problem. Glove uses the co-occurrence matrix to represent the local and global information. It constructs a word-to-word matrix and obtains the word representation by decomposing the matrix. The decomposition method is borrowed from the method of Latent Factor Model in the recommendation system.

Both Word2vec and Glove have a common problem. The same word has different semantics in different contexts but the embedding representation of this word in different contexts are same. ELMo proposed by Peters [11] optimizes this problem. ELMo has ability to learn the complicated features of morphology and the changes in different contexts.

There is a problem in model ELMo. The language model of ELMo is LSTM [17]. LSTM is unidirectional and infers information in order. So it cannot consider the data in other direction. LSTM [17] is a sequence model so the next step of processing cannot be calculated until the previous step is completed. Google proposes a model BERT [3] after ELMo. It uses transformer [19] to encode and considers the contextual features in both directions when predicting words. BERT shares a very similar idea with ELMo and performs better than ELMo.

In this paper, we choose static word embedding model Glove [10]. And there is a word embedding which combines with entity and link information in Wikipedia to get specific embeddings of entities. So we also choose Wikipedia2Vec [22] word embedding to compare with Glove.

2.3 Open-world knowledge graph model

Different from knowledge graph representation learning model, there are only few other works on open-world knowledge graph. Xie proposes a model named Description-Embodied Knowledge Representation Learning (DKRL) [21]. DKRL utilizes a joint training for KG embedding (TransE) and word embedding. It develops two different encoders to encode descriptive text of entity, including deep convolutional neural model and continuous bag-of-words. Shi and Weninger [15] propose a text-centric model ConMask. It obtains features from entity name, descriptive text of entity and relation. The score of triplets is calculated by fully convolutional networks [14]. ConMask need implicit reasoning based on the context, so the evaluation of link prediction depends on the length of descriptive text greatly.

In contrast to DKRL and ConMask, Haseeb Shah [13] proposes a new joint model OWE. It trains knowledge graph and word embedding of descriptive text independently. It uses the descriptive text of the entity which is not in the graph instead representation learning of this entity to calculate the score of the triplet. Haseeb Shah [13] produces a dataset FB15k-237-OWE in OWE for us to conduct open-world KG completion experiment.

In this paper, our model compares with the above three models on the open-world KG task.

3 Caps-OWKG: a capsule network model for open-world knowledge graph

we introduce the structure details of our proposed model and the specific steps in application. We briefly introduce the model calculation process through the following example. The specifics of each process are then detailed.

Example task: Judge whether the triplet (*Australia national soccer team, Position, Midfielder*) is true.

Head entity description: “*sports team that represents Australia*”.

Tail entity description: “*association football position played on both ends of the field*”.

Probability of triplet authenticity: 0.90

We calculate the probability of whether this triplet is true. First, we choose a knowledge graph representation learning model that has been trained, and obtain embedding representations of head(tail) entities and relations in the triplets. These embedding representations can reflect the structural characteristics of the triplets. Secondly, what we need to deal with is the descriptive text of the head(tail) entities. We

utilize a pre-trained word embedding model to process the words in the text, and then adjust the embedding representations again based on the relationship between entities and relations in the triplets.

According to the above processing flow, our model has the following four components:

- Knowledge graph processing component, get structural embedding representation.
- Text processing component, which processes the descriptive text of the entities and obtains a descriptive vector embedding of the triplets.
- Fusion component, combines structural embedding and descriptive embedding.
- Capsule network component, which the fused embedding is processed to obtain the final triplet probability result.

Knowledge graph processing component, Text processing component and Fusion component are shown as Fig. 1. And we will introduce each part in detail.

3.1 Knowledge graph processing component

In this component, we need to process the information of the knowledge graph and extract some structural features of knowledge graph. We utilize the traditional knowledge graph representation learning model to get the embeddings of entities and relationships in knowledge graph. There are many traditional representation learning models, and many have already performed very well in tasks such as relationship prediction. The embedding representation of the graph is the most basic part of the entire model. Other open-world knowledge graph models have been expanded and evolved based on this stage. Related models such as ConMask [15]

and TransE-OWE [13] have chosen the classic model TransE [1] at this stage. The capsule network which we will use is also a follow-up experiment based on TransE. Therefore, in order to compare with other models, we also choose the TransE model at this stage as the representation learning model of the knowledge graph processing component. When an entity is unseen in the knowledge graph, we use zero vector to instead its structural embedding.

As for a triplet (h, r, t) , we use pre-trained model TransE as function:

$$f_{\text{structural}}((h, r, t)) = (h_s, r_s, t_s) \quad (2)$$

The structural embedding presentation of triplet is (h_s, r_s, t_s) .

3.2 Text processing component

In this component, we process the descriptive text of the entity, and finally obtain an embedding representation to express the descriptive characteristics of the entity. We first need to get embedding of the words in the text, and then proceed to the next processing. In this step, we choose two different word embedding representation models for comparison. They are Glove embedding [10] and Wikipedia2Vec embedding [22].

In the representation step, for the entity's name, we need filter whether the representations of phrases are in the model (thesaurus). If a phrase is not in the model (thesaurus), the phrase is split into words for embedding representation. During the representation conversion process, if a word that does not appear in the model (thesaurus), we use zero vector instead.

Completing the embedding replacement of each word in the text is only the first step, and then the relationship

Fig. 1 Knowledge graph processing component, text processing component and fusion component of the model

strengthening process is to express the characteristics of the triplets. We match the descriptive text of each entity with the relation embedding of the triplets. The embedding representation of the relation is determined by the relation's name. The word embedding model is utilized to get the relation's name embedding. The same as entity's name, if the phrase of name is in the model (thesaurus), it is directly represented by an embedding, otherwise the relation's name is split into words. Then the mean of word embeddings is used to represent the relation's name.

$$r(w_i, r_e) = \frac{w_i \cdot r_e}{|w_i| |r_e|} \quad (3)$$

where w_i denotes as the embedding of the i -th word in entity, and r_e is the embedding of the relation's name. $r(w_i, r_e)$ calculates the correlation between the relation's name and each word in the descriptive text. Then we utilize the correlation to get the weight u_i .

$$u_i = \frac{e^{r(w_i, r_e)}}{\sum_{i=1}^k e^{r(w_i, r_e)}} \quad (4)$$

where k denotes as the number of word in entity's name and descriptive text. Then we add the text descriptive words embedding as the descriptive embedding representation of the entity.

$$v_{\text{Entity}} = \frac{\sum_{i=1}^k u_i w_i}{k} \quad (5)$$

This step can highlight the relationship between entities and relation in the triplets. This makes the descriptive embedding of the same entity in different triplets. It highlight the specificity of the entity in different contexts.

The descriptive embedding presentation of triplet is (h_d, r_d, t_d) .

3.3 Fusion component

In this component, we fuse the structural embedding presentation h_s, r_s, t_s of triplet and the descriptive embedding presentation (h_d, r_d, t_d) of triplet into a new embedding representation (h_{sd}, r_{sd}, t_{sd}) to express the triplets. This fusion process is divided into two steps. The first step is mapping from structural embedding vector space to descriptive embedding vector space. This can avoid errors caused by different spatial fusion. In the second step, after the vector space is unified, the two embedding in the same space are fused. In this step, we use two different fusion methods for comparison.

Mapping: We use three forms of mapping functions for comparison. They are as follows:

- 1) Proportional Function: A simple directly proportional function $f^{\text{map}}(x) = A \cdot x$.
- 2) Linear Function: Here, f^{map} denotes as a linear function $f^{\text{map}}(x) = C \cdot x + D$.
- 3) Multi-Layer Perceptron: f^{map} denotes as a several layer Multi-Layer Perceptron (MLP) with ReLU activation functions.

To train the mapping function, we use the structural embedding presentation h_s, r_s, t_s and the descriptive embedding presentation (h_d, r_d, t_d) . Finally, f^{map} is trained by minimizing the loss function:

$$L(\theta) = \sum_{i=1}^n (||f^{\text{map}}(h_{d_i}) - h_{s_i}||_2 + ||f^{\text{map}}(r_{d_i}) - r_{s_i}||_2 + ||f^{\text{map}}(t_{d_i}) - t_{s_i}||_2) \quad (6)$$

using stochastic gradient descent, where n denotes as the amount of triplet in train set and θ denotes as the parameters of f^{map} .

Fusion: After the vector space mapping is completed, we fuse two vector tuples of the same triplet. We try to use two different fusion methods for comparison:

- 1) Splice: We splice two embedding directly $h_{sd} = (s_1, s_2, \dots, s_{dim}, d_1, d_2, \dots, d_{dim})$, where $h_s = (s_1, s_2, \dots, s_{dim})$ and $h_d = (d_1, d_2, \dots, d_{dim})$.
- 2) Mean: Here, we calculate the mean of two embedding $h_{sd} = \frac{h_s + h_d}{2}$.

The result of the final fusion becomes the final representation embedding (h_{sd}, r_{sd}, t_{sd}) of the triplets.

3.4 Capsule network component

Capsule network [7] perform well in close-world knowledge graph tasks. We also use similar settings here to handle open-world knowledge graph task. We compare and investigate whether the capsule network can perform the same excellent performance for open-world knowledge graph. For processed triplets, we take a similar approach to static networks.

In our Capsule network component, we follow the paper [13] to view each embedding triplet (h_{sd}, r_{sd}, t_{sd}) as a matrix $A = [h_{sd}, r_{sd}, t_{sd}] \in R^{k \times 3}$, and $A_{i,:} \in R^{1 \times 3}$ is the i -th row in A . We utilize a filter $\omega \in R^{1 \times 3}$ which is processed in convolution layer. This filter ω is processed by every row in A to produce a feature map $q = [q_1, q_2, \dots, q_k] \in R^k$, in which $q_i = g(\omega \cdot A_{i,:} + b)$ where $\cdot$ denotes a dot product, g represents a non-linear activation function and $b \in R$ represents a bias term. Generate feature maps are generated by multiple filters $\in R^{1 \times 3}$. In our model, $N = |\Omega|$ is the number of filters and Ω is the set of filters. So we can obtain N k -dimensional

feature maps, for which every feature map catches a characteristic among entries at the same dimension.

We simplify the architecture by the capsule network which is built with two single capsule layers. We build k capsules and encapsulate the entries which are from the same dimension in all feature maps into a homologous capsule in 1st layer. So, every capsule is able to get lots of features of the entries which are from the homologous dimension in a triplet. In 2nd layer, we generalize features into a capsule and obtain an output. And the length of output embedding is the score of the triplet.

There are k capsules in 1st capsule layer and every capsule $i \in \{1, 2, \dots, k\}$ has an output embedding $u_i \in R^{N \times 1}$. Weight matrices $W_i \in R^{d \times N}$ multiply output embeddings

u_i to obtain embeddings ($\hat{u}_i \in R^{d \times 1}$ which are summed to produce an input embedding $s \in R^{d \times 1}$ for 2nd layer capsule. Then the capsule produces an output embedding $e \in R^{d \times 1}$ by the non-linear squashing function:

$$e = \text{squash}(s); s = \sum_i c_i \hat{u}_i; \hat{u}_i = W_i u_i \quad (7)$$

where $\text{squash}(s) = \frac{\|s\|^2}{1 + \|s\|^2} \frac{s}{\|s\|}$, and c_i are coupling coefficients which are defined by the routing process. The routing process is described in Algorithm 1 in detail. In 2nd layer there is only one capsule, we apply the softmax in a direction from all capsules in the previous layer to each of capsules in the next layer.

Algorithm 1 The routing process proposed by Sabour. [12]

```

1: for all capsule  $i \in 1st\ layer$  do
2:    $b_i \leftarrow 0$ 
3: end for
4: for iteration = 1, 2, ...,  $m$  do
5:    $c \leftarrow \text{softmax}(b)$ 
6:    $s \leftarrow \sum_i c_i \hat{u}_i$ 
7:    $e = \text{squash}(s)$ 
8: end for
9: for all capsule  $i \in 1st\ layer$  do
10:   $b_i \leftarrow b_i + \hat{u}_i \cdot e$ 
11: end for

```

We introduce our model in Fig. 2 with the embedding size: $k = 5$, the amount of filters: $N = 6$, the amount of neurons in 1st layer is equal to N , and the number of neurons within the capsule in 2nd layer: $d = 3$. The length of the output embedding e is the score for the input triplet.

The score function f of the triplet (h, r, t) is shown as follows:

$$f(h, r, t) = \|\text{capsnet}(g([(h_{sd}, r_{sd}, t_{sd})] * \Omega))\| \quad (8)$$

where the set of filters Ω is shared parameters in the convolution layer; $*$ is a convolution operator; and capsnet is a

capsule network operator. We utilize the Adam optimizer to train Capsule network by minimizing the loss function [6, 18]:

$$L = \sum_{(h,r,t) \in \{E \cup E'\}} \log(e^{-t_{(h,r,t)} f(h,r,t)} + 1) \quad (9)$$

where $t_{(h,r,t)} = \begin{cases} 1, & \text{if } (h, r, t) \in E \\ -1, & \text{if } (h, r, t) \in E' \end{cases}$, here E is a valid set and E' is an invalid set. E' is introduced by corrupting valid triplets in E .

Fig. 2 Our Capsule network example $k = 5, N = 6, d = 3$

4 Experiments

To verify the effectiveness of our model, we perform experiments on the common datasets for open-world knowledge graph task. Our model get the probability of the triplet authenticity, and we can use this probability to conduct the open-world knowledge graph link prediction task. So we use this experiment to compare with other open-world algorithms.

4.1 Datasets

For open-world knowledge graph link prediction task, the common datasets include DBPedia50k and FB15k-237-OWE. DBPedia50k was introduced by Shi and Weninger [15]. The average length of the descriptive text in DBPedia50k is 454 words. There is a lot of redundancy information in the descriptive text of this dataset. FB15k-237-OWE was introduced by Haseeb Shah [13]. FB15k-237-OWE is produced based on FB15K-237. To avoid the redundancy information, it filtered the descriptive text from Wikidata descriptions. It compressed the long text description into short text. The average length of the descriptive text in this dataset is only 5 words.

In the following experiment, we trained our model by triplets in the training set, then fine-tuned on the valid set, and finally made predictions in the test set. A brief introduction of the dataset is shown in the following Table 1.

Table 1 The statistics of dataset. E^{open} is the set of the entities which are not in KG

Dataset	$ E $	$ R $	$ E^{\text{open}} $	Number of triplets				
				Train	Head Pred.		Tail Pred.	
					Valid	Test	Valid	Test
DBPedia50k	24,624	351	3,636	32,388	55	2,139	164	4,320
FB15k-237-OWE	12,324	235	2,081	242,489	1,539	13,857	9,424	22,393

Table 2 Comparison with other models for tail prediction

Model	DBPedia50k				FB15k-237-OWE			
	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR
DKRL	–	–	40.0	23.0	–	–	–	–
ConMask	47.1	<u>64.5</u>	81.0	58.4	21.5	39.9	45.8	29.9
TransE-OWE	46.1	59.0	70.1	54.8	23.2	35.1	46.6	31.5
Caps-OWKG-200	<u>48.3</u>	63.6	74.9	<u>58.8</u>	<u>24.4</u>	38.1	<u>49.8</u>	<u>34.3</u>
Caps-OWKG-300	48.8	64.8	<u>75.8</u>	59.6	25.5	<u>39.0</u>	50.8	35.2

The best score denotes as bold, and the second best score denotes as underline

4.2 Settings

We get the parameters by training in the train set. The dimension of the knowledge graph pre-trained embedding is 200. The word embedding of descriptive text used is the pre-trained 200 dimension Glove embedding. The mapping function which we used is linear function and we used mean to fusion descriptive embedding and structural embedding. For our capsule network, the batch size is 128, the amount of neurons within the capsule in 2nd capsule layer is 10 ($d = 10$), and the amount of iterations m is in 1,3,5,7. We train capsule up to 200 epochs and record the Hits@10 score after each 20 training epochs to select the best hyper-parameters. The highest Hits@10 on the validation set are got by selecting $m = 1$, $N = 500$ and the initial learning rate $1e^{-5}$ at on DBPedia50k and $m = 1$, $N = 250$ and the initial learning rate $1e^{-5}$ at on FB15k-237-OWE.

Performance Measures Performance figures are computed using tail prediction on the test sets: For each test triplet ($head, relation, tail$) with open-world head $head \notin E$, we rank all known entities $tail' \in E$ by their score $f(h, r, t')$. We then evaluate the ranks of the target entities with the commonly used mean rank (MR), mean reciprocal rank (MRR), as well as Hits@1, Hits@3, and Hits@10.

4.3 Comparison result with other models

We use our model to compare with other models, including DKRL [21], ConMask [15] and TransE-OWE [13]. To get

a fair comparison, all results are test in the same test set. For all models, the basic knowledge graph representation learning model is TransE. In DKRL, ConMask, TransE-OWE and Caps-OWKG-200, word embedding used is 200 dimensional Glove embeddings. But in Caps-OWKG-300, we use 300 dimension Wikipedia2Vec embeddings to be word embedding of descriptive text.

The results of DKRL, ConMask are provided by Haseeb Shah [13]. DKRL is a two-layer CNN model that generates entity embeddings with entity description [21] and Baoxu Shi removed the structural-related features to make model work under open-world KGC settings [15]. ConMask uses a FCN over the entity names and descriptions along with the relation-based attention weights and the word embeddings in it are from the publicly available pre-trained 200 dimension Glove embeddings. And the result of TransE-OWE is calculated by us which use the code provided by Haseeb Shah [13] in Github. TransE-OWE utilizes TransE to obtain the structural embeddings representation of the KG and uses averaging as an aggregation function to obtain descriptive embeddings of entities.

In Table 2, we can see that our model has better results than others. When all models used the same word embedding model, Caps-OWKG-200 has the best results in MRR and H@1 of two datasets. We believe that our model can fusion the descriptive text very well and capsule network in our model can extract and process the features of triplets again. So our model performs better than others for open-world link prediction task. The results of the experiment verified that capsule networks also perform well on open-world knowledge graph. Both our model and TransE-OWE are used the similar method to link the structural embeddings with the descriptive embeddings. Compared with TransE-OWE, our model optimizes the link method and further processes the fused embeddings of triplets. So the experimental results on two datasets of our model are better than those of TransE-OWE in the four evaluation indicators. In the evaluation indicators Hits@10 and Hits@3, the results of ConMask are very excellent. Furthermore, the result on dataset DBPedia50k in the evaluation indicators Hits@10 is better than our model. It means that for prediction entities the recall of ConMask is better than our model but the precise is worse than ours. In our opinion, it is because of the fused method of entity names, relation name and entity descriptions. On dataset FB15k-237-OWE the results of our model are better than ConMask significantly. We believe that it is because of FB15k-237-OWE having very short descriptions. ConMask generally relies on extracting information from the description of entities with its attention mechanism, whereas our model relies more on extracting information from the textual corpus that the word embedding were trained on. This enables our model to provide good results without relying on having long descriptions.

Fig. 3 Performance when removing the entity descriptions

Table 3 Comparison of different map function with Caps-OWKG-300 on the FB15k-237-OWE dataset

Map function	MRR	Hits@1	Hits@3	Hits@10
Proportional	32.8	24.6	37.4	47.7
MLP	35.3	25.6	38.6	50.7
Linear	35.2	25.5	39.0	50.8

The best score denotes as bold

4.4 Analysis of different word embedding and effect of descriptive text

In text processing component, we choose two different word embedding model to process the descriptive text. We compare two models on dataset FB15k-237-OWE. In Figure 3 and Table 2, we can see that the performance of Caps-OWKG-300 which used 300 dimension Wikipedia2Vec embedding better than Caps-OWKG-200 which used 200 dimension Glove embedding. Wikipedia2Vec embedding contain phrase embedding and it can have better presentation of relation's name than Glove. So in the relationship strengthening process of text processing component, Wikipedia2Vec embedding can perform better than Glove.

In some cases, the descriptive text may miss or lack some information. Open-world models are all used descriptive text and so is our model. We do experiments to simulate the lack of text information and analyze the importance of text information. We randomly remove only the descriptive text for different percentages of chose entities (from 0% to 100%). The entities' names are not removed.

In Fig. 3, we can find that as the text information decreases, the results of prediction continue to decline. It confirms the importance of the descriptive text in

Table 4 Comparison of different fusion method with Caps-OWKG-300 on the FB15k-237-OWE dataset

Fusion method	MRR	Hits@1	Hits@3	Hits@10
Splice	32.5	24.5	37.1	47.4
Mean	35.2	25.5	39.0	50.8

The best score denotes as bold

Table 5 Hits@10 on the FB15k-237-OWE with $N = 150$ and the initial learning rate at $1e^{-5}$

m	20	40	60	80	100
1	49.03	50.02	50.55	50.81	50.80
3	47.96	49.35	49.87	50.02	50.11
5	47.00	49.30	49.03	49.12	49.16
7	41.34	43.54	43.11	43.35	43.44

The best score denotes as bold

open-world model. And Glove are more affected than Wikipedia2Vec. Wikipedia2Vec embedding are trained so we can retrieve entity embedding by the name alone.

4.5 Analysis of different mapping function and fusion

In the fusion component, we use three different mapping functions and two different fusion methods. In Table 3, we can see that our model uses linear mapping function performs better than proportional function. The accuracy of linear function is 3% higher than proportional function. The results of Multi-Layer Perceptron and Linear function are very close. But training Multi-Layer Perceptron function need more time cost than Linear function. So considering the time cost and accuracy, we believe that linear function is more suitable than others.

In Table 4, we can find that the model uses mean method to fusion structural embedding and descriptive embedding performs better than splicing method to fusion. The accuracy of mean method is 3% higher than splicing method. It is because that when an entity is unseen in the knowledge graph, splicing and mean method use zero vector to instead. But zero vector in splicing method can cause a bigger impact than it in the mean method. So, for unseen entity, it is available for us to use mean method to fusion two kinds of embedding.

4.6 Effects of routing iterations in capsule network

In this part, we analyze the impact of routing iterations on performance. Table 5 exhibits the Hits@10 on the FB15k-237-OWE for a comparison with respect to each value of the routing iterations with $N = 150$ and the Adam initial

learning rate at $1e^{-5}$. We can find that the best performance is got by setting $m=1$. This setting is different from the setting in the task of image classification and m is set greater than 1 in the image classification task. It can obtain the relative positions of entities in the image (such as eyes, nose and mouth) properly. But in knowledge graph, there are some different relations between the same entity couple (head entity and tail entity). So the setting which the number m of routing iterations is greater than 1 may be not right for the 1-to-N, N-to-1 and N-to-N relations in the knowledge graphs. In our model, we choose $m=1$ to be our general setting.

5 Conclusions and future work

In this paper, a new model based on the capsule networks for the open-world knowledge graph task is proposed. Our model predicts the authenticity of triplets in knowledge graph. We use the probability of triplet authenticity to conduct the open-world knowledge graph link prediction task. It gets very excellent results on two common datasets FB15k-237-OWE and DBPedia50k. In text processing component, we used relationship strengthening process to represent the descriptive embedding of triplets better. After comparing several mapping functions and fusion methods, we chose the most available function and method for our model.

The difference between our model and other methods is that, embedding representation of triplets can be regarded as dynamic embedding presentation in the application. In future work, an interesting direction is trying dynamic embedding to predict open-word knowledge graph link. To get better dynamic embedding of triplets, we can also use dynamic word embedding to instead Wikipedia2Vec used in our model in the future. The basic representation learning model we used in our model is TransE. But it is not the best model for representation learning. We can try other models to combine with capsule network in the future.

Acknowledgements This work was supported by NSFC under grants Nos. 61872446, 61902417 and 71971212, NSF of Hunan Province under grant No. 2019JJ20024.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in

the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bordes A, Usunier N, García-Durán A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. In: Burges CJC, Bottou L, Ghahramani Z, Weinberger KQ (eds) *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*. Proceedings of a meeting held December 5–8, 2013. Lake Tahoe, Nevada, United States, pp 2787–2795
- Bordes A, Glorot X, Weston J, Bengio Y (2014) A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. *Mach Learn* 94(2):233–259. <https://doi.org/10.1007/s10994-013-5363-6>
- Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) *Proceedings of the 2019 Conference of the North American Chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2019*. Association for Computational Linguistics, Stroudsburg, pp 4171–4186. <https://doi.org/10.18653/v1/n19-1423>
- Lin Y, Liu Z, Sun M, Liu Y, Zhu X (2015) Learning entity and relation embeddings for knowledge graph completion. In: Bonet B, Koenig S (eds) *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, Austin, pp 2181–2187
- Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: Bengio Y, LeCun Y (eds) *1st International Conference on Learning Representations, ICLR 2013*, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings
- Nguyen DQ, Nguyen TD, Nguyen DQ, Phung DQ (2018) A novel embedding model for knowledge base completion based on convolutional neural network. In: Walker MA, Ji H, Stent A (eds) *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. Association for Computational Linguistics, Stroudsburg, pp 327–333. <https://doi.org/10.18653/v1/n18-2053>
- Nguyen DQ, Vu T, Nguyen TD, Nguyen DQ, Phung DQ (2019) A capsule network-based embedding model for knowledge graph completion and search personalization. In: Burstein J, Doran C, Solorio T (eds) *Proceedings of the 2019 Conference of the North American Chapter of the association for computational linguistics: human language Technologies, NAACL-HLT 2019*. Association for Computational Linguistics, Stroudsburg, pp 2180–2189. <https://doi.org/10.18653/v1/n19-1226>
- Nickel M, Tresp V, Krieger H (2011) A three-way model for collective learning on multi-relational data. In: Getoor L, Scheffer T (eds) *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*. Omnipress, Bellevue, pp 809–816
- Nickel M, Rosasco L, Poggio TA (2016) Holographic embeddings of knowledge graphs. In: Schuurmans D, Wellman MP (eds) *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, Phoenix, pp 1955–1961
- Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Moschitti A, Pang B, Daelemans W (eds) *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, Doha, pp 1532–1543. <https://doi.org/10.3115/v1/d14-1162>
- Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Walker MA, Ji H, Stent A (eds) *Proceedings of the 2018 Conference of the North American Chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2018*. Association for Computational Linguistics, Stroudsburg, pp 2227–2237. <https://doi.org/10.18653/v1/n18-1202>
- Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 4–9 December 2017. Long Beach, CA, USA, pp 3856–3866
- Shah H, Villmow J, Ulges A, Schwanecke U, Shafait F (2019) An open-world extension to knowledge graph completion models. *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*. AAAI Press, London, pp 3044–3051. <https://doi.org/10.1609/aaai.v33i01.33>
- Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(4):640–651. <https://doi.org/10.1109/TPAMI.2016.2572683>
- Shi B, Weininger T (2018) Open-world knowledge graph completion. In: *The thirty-second AAAI conference on artificial intelligence (AAAI-18)*, pp 1957–1964
- Socher R, Chen D, Manning CD, Ng AY (2013) Reasoning with neural tensor networks for knowledge base completion. In: Burges CJC, Bottou L, Ghahramani Z, Weinberger KQ (eds) *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*. Proceedings of a meeting held December 5–8, 2013. Lake Tahoe, Nevada, United States, pp 926–934
- Sundermeyer M, Schlüter R, Ney H (2012) LSTM neural networks for language modeling. In: *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association*, Portland, Oregon, USA, September 9–13, 2012, ISCA, pp 194–197
- Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G (2016) Complex embeddings for simple link prediction. In: Balcan M, Weinberger KQ (eds) *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, New York City, NY, USA, June 19–24, 2016, JMLR.org, JMLR Workshop and Conference Proceedings, vol 48, pp 2071–2080
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 4–9 December 2017. Long Beach, CA, USA, pp 5998–6008
- Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph embedding by translating on hyperplanes. In: Brodley CE, Stone P (eds) *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, Québec City, pp 1112–1119
- Xie R, Liu Z, Jia J, Luan H, Sun M (2016) Representation learning of knowledge graphs with entity descriptions. In: Schuurmans D, Wellman MP (eds) *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, Phoenix, pp 2659–2665

22. Yamada I, Shindo H, Takeda H, Takefuji Y (2016) Joint learning of the embedding of words and entities for named entity disambiguation. In: Goldberg Y, Riezler S (eds) Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016. ACL, Berlin, pp 250–259. <https://doi.org/10.18653/v1/k16-1025>
23. Yang B, Yih W, He X, Gao J, Deng L (2015) Embedding entities and relations for learning and inference in knowledge bases. In: Bengio Y, LeCun Y (eds) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.