Representation Learning of Knowledge Graphs with Entity Attributes and Multimedia Descriptions

Yukun Zuo*, Quan Fang†, Shengsheng Qian†, Xiaorui Zhang‡, Changsheng Xu†
*Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China
†National Lab of Pattern Recognition, Institute of Automation Chinese Academy of Sciences, Beijing, China
‡International School E-Commerce Engineering with Law, Beijing University of Posts and Telecommunications, Beijing, China
zykpy@mail.ustc.edu.cn {qfang, shengsheng.qian, csxu}@nlpr.ia.ac.cn zhangxiaorui@bupt.edu.cn

***Abstract*—Representation learning of knowledge graphs encodes both entities and relations into a continuous low-dimensional space. Most existing methods focus on learning representations with structured fact triples indicating relations between entities, ignoring rich additional information of entities including entity attributes and associated multimodal content descriptions. In this paper, we propose a new model to learn knowledge representations with entity attributes and multimedia descriptions (KR-AMD). Specifically, we construct three triple encoders to obtain structure-based entity representation, attribute-based entity representation and multimedia content-based entity representation, and finally generate the knowledge representations for knowledge graphs in KR-AMD. The experimental results show that, by special modeling of entity attributes and text-image descriptions, KR-AMD can significantly outperform state-of-the-art KR models in prediction of entities, attributes and relations, which validates the effectiveness of KR-AMD.**

***Index Terms*—Knowledge Graph, Multimodal, Embedding, Entity attributes**

I. Introduction

Knowledge graphs (KGs), which store complex structured information about the facts of the real world, have been successfully used in question answering [1], information extraction [2] and other fields. Typical knowledge graph such as Freebase [3], YAGO [4], and NELL [5] consists of entities (nodes) and relations (edges). Each edge of the graph represents a triple (head entity, relation, tail entity).

Recently, translation-based methods are proposed to project relations and entities of knowledge graph to low-dimensional semantic space, alleviating the computation and sparsity issues of traditional symbol-based representations. The most classic model is TransE [6], which considers relations to be translating operations between head and tail entities and shows promising effectiveness and efficiency in knowledge representation learning (KRL). However, most existing methods [6], [7] on KRL only concentrate on the structured information in fact triples of KGs, and largely ignore the rich external information of entities. In the real-world, the entities in knowledge graphs are always associated with descriptive attributes and multimodal content in addition to the relational structures.

In this paper, we focus on the knowledge representation learning problem, and propose a novel model to learn knowledge representations with entity attributes and multimedia descriptions (KR-AMD). For multimedia descriptions of each entity, we utilize image-based relational triple encoder and description-based relational triple encoder to obtain image-based vector representations and description-based vector representations, respectively and integrate them to the learning process of relational triples. In regard to the attributional triples of knowledge graph, we use attributional triple encoder to obtain attribute-based vector representations. Considering entity attributes and multimedia descriptions of knowledge graph together, as a result, the proposed model can effectively combine structure-based entity representation, attribute-based entity representation and content-based entity representation to learn knowledge representations for knowledge graphs.

We build a new real-world dataset FB55K and evaluate our model in three tasks: entity prediction, relation prediction and attribute prediction. The results show that our model KR-AMD can achieve better results than other state-of-the-art KR models by integrating the rich external information of the entity into KG representation, which demonstrates the effectiveness of our model.

II. Related Work

Translation-based methods have been adopted by many researchers in knowledge representation learning. TransE [6] projects entities and relations of triples into low-dimensional semantic vectors, and makes relations as translating operations between head entities and tail entities. TransE works well in 1-to-1 relations, but can not solve 1-to-N relations or N-to-1 relations effectively. To surmount this problem, TransH [7] defines a hyperplane for each relation $\mathbf{r}$. First, it projects the entity representations onto the hyperplane and then defines the scoring function similar to TransE. TransR [8] introduces relation-specific spaces, first projecting the entity representations vector into the space of the relation $\mathbf{r}$, and then defining the scoring function like TransE. TranA [9] obtains a symmetric non-negative matrix M for each relation $\mathbf{r}$, and then defines the scoring function with the adaptive Mahalanobis distance.

Entity external information such as textual and visual information is important for knowledge representation learning. [10] divides the relations of knowledge graph into relations and attributes. [11] projects its associated entity and text information into a common vector space with alignment model.

978-1-5386-5321-0/18/$31.00

[12] integrates visual information and textual information into a question-answer system. [13] integrates visual information into the knowledge graph. However, the combination of entity attributes and its associated multimedia descriptions has not been explored. Visual information and textual information have never been merged together into knowledge graphs based on the distinction of relations and attributes. Our model can fully integrate the entity multimedia information and affiliated attributes in a principled way.

III. Problem Formulation

Before explaining the details of our proposed model, we first introduce the notations used in this paper. Our model consists of two kinds of triples — $(h, r, t) \in T$ and $(h, a, v) \in Y$ while $h, t \in E$ represent entities, $r \in R$ depicts relation, $a \in A$ describes attributes and $v \in V$ stands for values. T contains the whole relational triples and Y represent all the attributional triples. E, R, A and V denote the set of entities, relations, attributes and values, respectively. To utilize the rich associated information of entities, we consider three kinds of representations as follows:

Definition 1. Structure-based Entity Representations: We set $\mathbf{h}_S$, $\mathbf{t}_S$ as the structure-based vector representations of head and tail entities, which are learned from the intrinsic information of knowledge graph.

Definition 2. Attribute-based Entity Representations: We set $\mathbf{h}_A$, $\mathbf{t}_A$ as the attribute-based vector representations of head entities and tail entities (attribute values), which are constructed from the attributional information of knowledge graph.

Definition 3. Content-based Entity Representation: We set $\mathbf{h}_D$, $\mathbf{t}_D$ as the description-based vector representations of head and tail entities, denote $\mathbf{h}_I$, $\mathbf{t}_I$ as the image-based vector representations of head and tail entities as well. These vector representations are built from the textual descriptions and images of entities, respectively.

IV. Methodology

To utilize both fact triples and additional entity information including attributes and text-image content information, our model KR-AMD is divided into three central components: (1) **Image-based Relational Triple Encoder** learns correlations between entities and relations from abundant visual information. (2) **Description-based Relational Triple Encoder** learns correlations between entities and relations from the descriptions of entities. (3) **Attributional Triple Encoder** embeds the correlations between entities, attributes and values.

A. Overall Architecture

The overall framework of KR-AMD is shown in Figure 1. The relations between triples are divided into relations and attributions. For each relational triple, we make full use of the visual image information and textual description information of head entity and tail entity. Encoding the images which corresponds to the entity to get image feature representations and projecting them to entity vector space, then we will get the

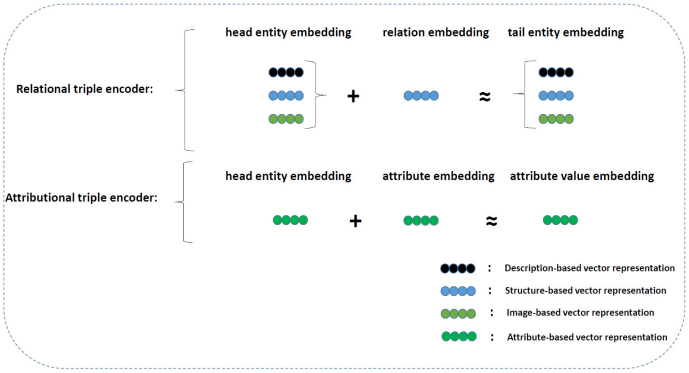

Fig. 1. The overall framework of KR-AMD

image-based vector representation using the attention mechanism. The textual descriptions information which corresponds to the entity is encoded by a deep convolution neural network to obtain a description-based vector representation. The structure-based vector representations, the obtained image-based vector representations and description-based vector representations are jointly trained to obtain ultimate relational triple representations. For each attributional triple, we learn individually and treat it as a classification problem. The overall energy function is defined as:

$$E = E_S + E_C + E_A$$

where $E_S = ||\mathbf{h}_S + \mathbf{r} - \mathbf{t}_S||$ is the energy function of structure-based vector representations, $E_C = E_I + E_D$ depicts the energy function of content-based vector representations. Next, let us introduce those energy functions in detail.

B. Image-based Relational Triple Encoder

For integrating the image information into the knowledge representation, the energy function of image-based relational triple encoder is defined as:

$$E_I = E_{SI} + E_{IS} + E_{II}$$

Among them, $E_{II} = ||\mathbf{h}_I + \mathbf{r} - \mathbf{t}_I||$ is the energy function based on the image-based vector representations. The head entity and the tail entity vectors represent the image-based information of the entities. $E_{SI} = ||\mathbf{h}_S + \mathbf{r} - \mathbf{t}_I||$ and $E_{IS} = ||\mathbf{h}_I + \mathbf{r} - \mathbf{t}_S||$ aim to guarantee the structure-based vector representations and image-based vector representations could be projected into the same vector space. The framework of image-based entity representation is presented in Figure 2.

For each entity, there are a couple of associated images denoted as $\{I_1, I_2, I_3..I_k\}$. We use AlexNet for feature extraction, which is a classical convolution neural network proposed by [14] in 2012. It is composed of a five-layer convolution network and two full-connection layers and one layer of softmax. We represent the output of second full-connection layer as the feature of image with a dimension of 4096. After the image feature representations are obtained, the image

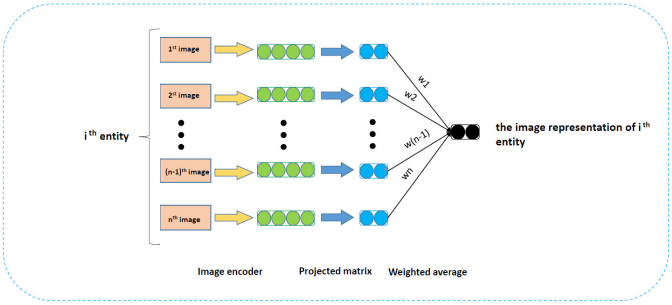

Fig. 2. Image-based entity representation.

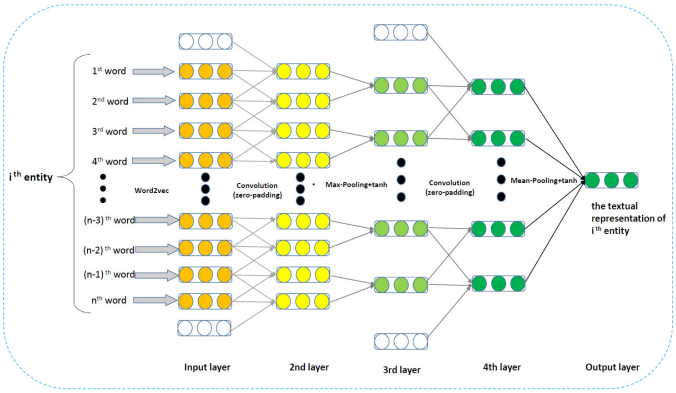

Fig. 3. Description-based entity representation.

feature representations are projected into the entity vector representation space using the projection matrix:

$$\mathbf{v}_i = \mathbf{M} \cdot f(I_i),$$

Each entity has multiple images. After obtaining the vectors of these images, the task has not been completed. We consider the image vectors and corresponding entity vectors to calculate the weights:

$$\mathbf{w}_i^{(k)} = \frac{\exp(\mathbf{v}_i^{(k)} \cdot \mathbf{q}_S^{(k)})}{\sum_{j=1}^{n} \exp(\mathbf{v}_j^{(k)}, \mathbf{q}_S^{(k)})}$$

$\mathbf{v}_i^{(k)}$ represents the i-th image vector of the k-th entity while $\mathbf{q}_S^{(k)}$ depicts the structure-based vector of the k-th entity. The denominator of the above formula means traversing the whole image vector of the k-th entity. We weight all image vectors and sum them to get the final image-based vector representation of the k-th entity:

$$\mathbf{q}_I^{(k)} = \sum_{i=1}^{n} \frac{att(\mathbf{v}_i^{(k)}, \mathbf{q}_S^{(k)}) \cdot \mathbf{v}_i^{(k)}}{\sum_{j=1}^{n} att(\mathbf{v}_i^{(k)}, \mathbf{q}_S^{(k)})}$$

C. Description-based Relational Triple Encoder

The energy function of description-based relational triple encoder is defined similarly to previous one:

$$E_D = E_{SD} + E_{DS} + E_{DD}$$

In the above formula, $E_{DD} = ||\mathbf{h}_D + \mathbf{r} - \mathbf{t}_D||$ is the energy function based on description-based vector representations of the entity. The head entity and the tail entity vectors contain the textual information of the entity. $E_{SD} = ||\mathbf{h}_S + \mathbf{r} - \mathbf{t}_D||$ and $E_{DS} = ||\mathbf{h}_D + \mathbf{r} - \mathbf{t}_S||$ assure that structure-based vector representations and description-based vector representations could be located into the same vector space. Figure 3 shows the flowchart of description-based entity representation.

For capturing the textual description information of the entities, we use a convolution neural network to process the textual descriptions. The textual descriptions of the entities directly adopt the textual description dataset collected by the [15], which preprocesses the raw texts into a set of words. Each word can be processed to word embedding using Word2vec [16]. All the word embeddings for each entity are the input of the convolution neural network layer, and the output of the convolution neural network is the description-based vector representation of entity.

There are five layers in this convolution neural network architecture. The input layer utilizes the whole word embeddings of each entity as its input. Each word embedding can fully represent the characteristic of the corresponding word. The second layer is a convolution layer. We use a zero-padding method to handle the input of convolution layer. The third layer is a pooling layer, we adopt n-max-pooling method to select the max value within a size n window:

$$\mathbf{y}_i^{(2)} = \max(\mathbf{x}_{n \cdot i}^{(1)}, \ldots, \mathbf{x}_{n \cdot (i+1)-1}^{(1)})$$

This method can select the most significative feature value to reduce the computational complexity and model complexity of convolution neural network. The fourth layer is still a convolution layer. It learns deeper features. The output layer is a pooling layer, and we apply mean-pooling to get the final result. That is to say, all the feature values of the window are averaged:

$$\mathbf{y}^{(3)} = \sum_{i=1,\ldots,m} \frac{\mathbf{x}_i^{(2)}}{m}$$

The CNN architecture can handle entity description with different length and get a fixed-length textual vector representation.

D. Attributional Triple Encoder

To exploit the additional attributes of entities for knowledge representation learning, we use a single-layer neural network to project the vector of the entity space into the attribute space, and then calculate the similarity between the transformed representation vector and the vector of the corresponding attribute value:

$$E_A = ||f(\mathbf{h}_A \mathbf{W}_a + \mathbf{a}) - \mathbf{t}_A||$$

In the above formula, f is the activation function, $\mathbf{W}_a$ is a projected matrix and $\mathbf{t}_A$ is the vector of attribute value.

E. Objective Formalization

We adopt a margin-based loss function as our objective for training:

$$L = \sum_{(a,b,c)\in T,Y} \sum_{(a',b',c')\in T',Y'} \max(\lambda + E(a,b,c) - E(a',b',c'), 0)$$

In the above formula, λ represents the margin hyperparameter and $E(a,b,c)$ is the energy functions discussed previously. T' is the wrong relational triple training set, which is defined as:

$$T' = (a',b,c)|a' \in E \ \cup \ (a,b,c')|c' \in E \cup (a,b',c)|b' \in R, \quad (a,b,c) \in T$$

$Y^{'}$ represents the wrong attributional triple training set, which is defined analogously:

$$Y' = (a,b,c')|c' \in R, \quad (a,b,c) \in Y$$

We randomly replace the head entity, tail entity or relation of the correct triple to get T', not including the correct relational triple in T. Similarly, Y' is constructed by replacing the attribute value of the correct attributional triple, ignoring the correct triple in Y.

F. Optimization and Implementation Details

Obviously, our KR-AMD model could be summarized as a parameter set θ =(**W**, **E**, **R**, **M**, **A**, **V**) where **E**, **R**, **A**, **V** represent the embedding set of entities, relations, attributions, attribute values, respectively. **W** stands for the weights of the convolution neural networks in description-based relational triple encoder, and **V** depicts the projection matrix of image-based relational triple encoder.

We use a stochastic gradient descent method as an optimization strategy. The word vector for each word is obtained by Word2vec [16]. The image feature representations for each image are calculated using the pre-trained AlexNet [14]. The parameters of the KR-AMD model are initialized randomly and we use a multi-thread version to train our model.

V. EXPERIMENTS

A. Dataset

We have constructed a new knowledge graph dataset named **FB55K**. Most of entities in the relational triples of this dataset have the corresponding textual description information and visual image information. In fact, the triples of the knowledge graph dataset FB55K are a subset of the knowledge graph dataset FB24K [10]. For the textual description information of the knowledge graph data set FB55K, we directly use the textual description set of entities collected in [15] . For visual image information, we crawl five pictures for each entity in the search engine Bing[1] as its visual information. The statistics of FB55K are listed in Table 1.

[1]https://cn.bing.com/images/trending?form=Z9LH

Table 1:Statistics of dataset

Dataset	FB55K
#Entities	17871
#Relations	429
#Attributes	314
#Train(Relational Triples)	78723
#Test(Relational Triples)	4147
#Train(Attributional Triples)	6663
#Test(Attributional Triples)	324

B. Baselines

According to our model, we divide the knowledge graph completion into three parts: entity prediction, relation prediction and attribute prediction. In each task, we replace the predicted part of the triple with the elements in the corresponding set and calculate the score of these triples then sort these triples. We set the classical methods TransE [6], KR-EAR [10], IKRL [13] and DKRL [15] as our baselines and implement these methods on our dataset. We also implement a model named KR-MD which only considers structure-based information and content-based information and ignores attribute-based information. By constrast, KR-EAR is the model which only considers structure-based information and attribute-based information by neglecting content-based information. Because the purpose of incorporating visual and textual information in the model is to better learn the embedding of the knowledge graph, we only use structure-based embeddings for the comparison among models.

C. Entity Prediction

Entity prediction refers to predict the correct entity of the relational triple when head entity or tail entity is missing. According to the paper [6], we use the two methods described in the article as our evaluation metrics: the average rank of the correct triple (Mean Rank) and the proportion of correct triples in top10 (Hits@10). We also follow two evaluation settings named "Raw", "Filter" introduced in [6].

Table2: Evaluation results on entity prediction

model	Mean Rank		Hits@10	
	Raw	Filter	Raw	Filter
TransE	292.718	259.164	0.3536	0.5412
IKRL	274.408	242.149	0.3603	0.5477
DKRL	266.233	233.727	0.3606	0.5450
KR-EAR	275.035	242.338	0.3588	0.5482
KR-MD	281.481	248.462	0.3589	0.5403
KR-AMD	**257.658**	**224.904**	**0.3635**	**0.5484**

The results of the entity prediction are shown in Table 2. From the results, we can clearly see that: (1) The KR-EAR utilizing entity attributes and KR-MD considering multimodal content of entities have better performances compared with the TransE model. (2) The proposed KR-AMD model significantly and consistently outperforms other KRL approaches. This observation shows that the quality of knowledge representation can be improved by considering relational facts, entity attributes, and entity multimedia descriptions simultaneously.

D. Relation Prediction

Relation prediction refers to predict the correct relation between two given entities. We use the two indicators: Mean Rank and Hits@1 to evaluate the quality of the model:

Table 3: Evaluation results on relation prediction

model	Mean Rank		Hits@1	
	Raw	Filter	Raw	Filter
TransE	2.693	2.426	0.7039	0.8734
IKRL	2.532	2.267	0.7029	0.8736
DKRL	2.565	2.297	0.6976	0.8756
KR-EAR	2.653	2.379	0.6969	0.8744
KR-MD	2.764	2.495	0.7031	0.8763
KR-AMD	**2.484**	**2.218**	**0.7046**	**0.8821**

The results of the relation prediction are shown in Table 3. From the table, we can see that: (1) The result of KR-AMD in Mean Rank is superior to KR-EAR, TransE, IKRL, DKRL and KR-MD. (2) In Hit@1, our model outperforms all other models. This demonstrates the effectiveness of considering entity attributes and content information in representation learning of knowledge graph for relation prediction.

E. Attribute Prediction

The goal of attribute prediction is to predict missing attribute values in attributional triples. We use the two measures Mean Rank and Hits@1 to evaluate the quality of the model:

Table4: Evaluation results on attribute prediction

model	Mean Rank		Hits@1	
	Raw	Filter	Raw	Filter
KR-EAR	8.083	7.4846	0.608	0.744
KR-AMD	**6.605**	**6.006**	**0.630**	**0.765**

The results of attribute prediction are shown in Table 4, because TransE, IKRL, KR-MD and DKRL do not involve attributes, so we only compare with KR-EAR model. From the Table 4, we can see that KR-AMD added entity attributes and text-image content information to the model has achieved better results than KR-EAR. It can be seen that although we train the embedding of attributional triples independently, the rich external information of entity can enhance the representation of the entity embedding significantly.

VI. Conclusion and future work

In this paper, we propose the KR-AMD model for representation learning of knowledge graphs with entity attributes and multimedia descriptions. From the experimental results, we can see that our model outperforms state-of-the-art models on all three sub-tasks by integrating the rich external information of entities.In the future, we will conduct further research from the following directions: (1) Integrating more kinds of information into knowledge graph. (2) We will make our model extend to knowledge inference, knowledge classification and other tasks.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 61432019, 61572498 , 61720106006 and 61702509, the Key Research Program of Frontier Sciences, CAS, Grant NO. QYZDJ-SSW-JSC039, and the Beijing Natural Science Foundation 4172062.

References

[1] Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 165–180, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[2] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124, New York, NY, USA, 2013. ACM.

[3] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, 2008.

[4] 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*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.

[5] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pages 1306–1313. AAAI Press, 2010.

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc., 2013.

[7] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes, 2014.

[8] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion, 2015.

[9] Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. Transa: An adaptive approach for knowledge graph embedding. *CoRR*, abs/1509.05490, 2015.

[10] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Knowledge representation learning with entities, attributes and relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2866–2872. AAAI Press, 2016.

[11] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In *EMNLP*, 2014.

[12] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, Dec 2015.

[13] Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Image-embodied knowledge representation learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 3140–3146. AAAI Press, 2017.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[15] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2659–2665. AAAI Press, 2016.

[16] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.