IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 18, 2019, accepted April 11, 2019, date of publication April 24, 2019, date of current version May 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913086

TDN: An Integrated Representation Learning Model of Knowledge Graphs

XIAOJUN KANG[1,2], HONG YAO[1,2], (Member, IEEE), QINGTAO LI[1], XINCHUAN LI[1,2], CHAO LIU[1,2], AND LIJUN DONG[1,2]

[1]School of Computer Science, China University of Geosciences, Wuhan 430074, China
[2]Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, China

Corresponding author: Hong Yao (yaohong@cug.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1004600, in part by the National Science and Technology Major Project of China under Grant 2017ZX05036-001-010, and in part by the National Natural Science Foundation of China (NSFC) under Grant 61672474, Grant 61501412, and Grant 61673354.

ABSTRACT Knowledge graph (KG) is playing an important role in many artificial intelligence applications. Representation learning of KGs aims to project both entities and relations into a continuous low-dimensional space. The representation learning technique based on embedding has been used to implement the KG completion, which aims to predict potential triples (head, relation, and tail) in KG. Most current methods concentrate on learning representations based on triple information while ignoring integrating the textual knowledge and network topology of KG. This leads to ambiguous completions. To address this problem and implement more accurate KG completion, we propose a new representation learning model, TDN model, which integratedly embeds the information of triples, text descriptions, and network structure of KG in a low-dimensional vector space. The framework of TDN is defined and the methodology of implementing TDN embedding is explored. To verify the effectiveness of the proposed model, we evaluate TDN via the experiments of link prediction on the real-world datasets. The experimental results confirm the above claims and show that TDN-based embedding significantly outperforms other baselines.

INDEX TERMS Artificial intelligence, embedding, knowledge graph, knowledge representation.

I. INTRODUCTION

Knowledge graph (KG) is playing an important role in many artificial intelligence (AI) applications such as intelligent question answering, web/mobile search, and semantic analysis, etc [1], [2]. KG is a semantic network consisting of a large number of triple facts like (*head*, *relation*, *tail*), where *head* and *tail* correspond the entities (nodes) in the network, and *relation* corresponds to the edge between *head* and *tail*. An important task of KG is completion (KG completion) which aims at predicting potential facts under the supervision of existing triples. Knowledge graph completion is similar to link prediction in social network analysis, but more challenging. Since KG includes varieties of symbolic and logical information, the link prediction needs to predict not only the existence of nodes but also the specific type and semantics of nodes. For this reason, traditional approach of link prediction is not capable for KG completion.

The associate editor coordinating the review of this manuscript and approving it for publication was Saqib Saeed.

To address this issue, the knowledge representation learning models based on *translation* (Trans), called *KG embeddings* [3]–[6] have been proposed, which attempt to embed components of a KG including entities and relations into continuous vector spaces [7] to quantitatively implement the accurate KG completion in the large-scale graphs [8]. In this paper, we call this mechanism *vector embedding*. Current knowledge representation learning models focus only on the triple-based information [3], while ignoring the integration of the textual knowledge based on entity descriptions and the topology information of network. This leads to ambiguous KG completions. Primarily, most of the knowledge graphs have some specific description texts for entities [9], and these descriptions contain important context information. The absence of context information cannot fine-grainedly analyze the relations in graphs for accurate reasoning, as the example shown in Figure 1(a). Furthermore, for a KG, all triples jointly compose a structured network with specific topology characteristics. In other words, any triple is not isolated but affected by others, and thus every entity has its own contribution to the

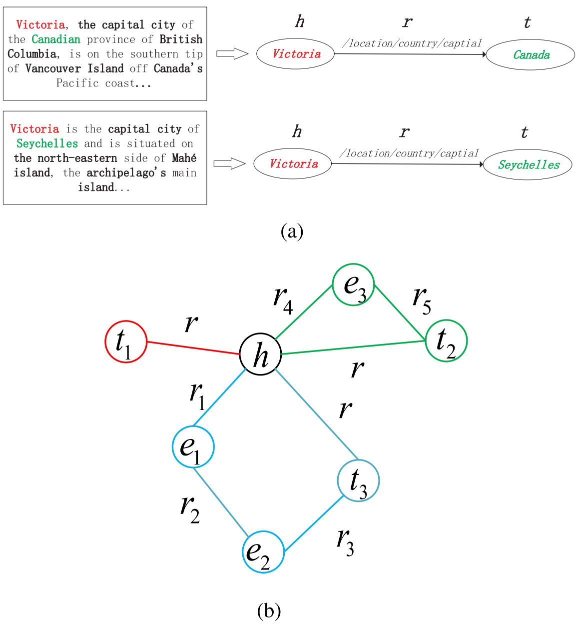


FIGURE 1. **The influence of entity descriptions and topology information on KG completion. (a) There are two head entities named "Victoria" with the same relation. According to the traditional Trans model, they would correspond to the same tail entity. However, with their respective text descriptions, they can be matched to different tail entities "Canada" and "Seychelles", respectively. (b) Three different tail entities (t_1, t_2, t_3) are matched to a head entity (h) by the same relation (r). Note that there are other different triples besides those triples tied with r, and every triple (h, r, t) in the KG has the contribution to the network topology rather than only brings its own structures. This should have helped to implement KG completion globally and precisely but has been ignored by the traditional triple-based models.**

network topology. But current models ignore this information when implementing embedding, as shown in Figure 1(b). As a result, the embedding with some information-loss will affect the accuracy of KG completion.

To address these problems, this study proposes a new representation learning model, called **TDN** model, to comprehensively handle the triple information (**T**), entity descriptions (**D**), and network structure (**N**). These features are integratedly embedded in a unified vector space, in which KG can be computed and analyzed with less ambiguity. The main contributions of this study are:

- An integrated representation model of KG is proposed, by which the triple information, entity descriptions, and network structure are considered jointly, and the KG can be represented with less ambiguity.
- By expressing the entities and relations with less ambiguity, KG completion can be implemented more accurately.

The reminder of this paper is organized as follows. We first introduce some related work and then profile the architecture of the TDN model. Next, the methodology for implementing TDN is proposed, and the trained model is presented. Furthermore, to evaluate the effectiveness of TDN, the KG completion experiments are conducted. Finally, our work is concluded and the future work is posed.

II. RELATED WORK

Current representation learning of KG can be broadly classified into two categories: the triple-based Trans embedding and the extra-information-based Trans embedding. The former utilizes the symbolic representation of triples only and the latter takes additional information (text information, image information, etc.) into representing the entities. As a classical Trans embedding based on triple information, TransE [3] embeds the entities and relations into a low-dimensional vector space by a translation process, by which every triple (h, r, t) is learned by the score function defined as

$$\mathbf{E}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|, \tag{1}$$

where $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$ is called the *embedding vectors* of h, r, t in this paper, respectively. TransE model is very simple and intuitive. However, because of embedding the entities and relations into the same vector space, its ability to handle the multi-relation (such as 1-to-N, N-to-1 and N-to-N relations) triples is limited [4], [5]. To overcome this drawback, TransH [4] has been proposed, which implements the embedding by a hyperplane of specific relation. TransH projects the head and tail onto the hyperplane and then completes the translation between entities. This model can make the same entities play different roles in different relation hyperplanes and implement more fine-grained translations for multi-relation cases. Furthermore, TransR [5] proposes a transformation matrix to separate entity space from relational space and utilizes the matrix to map entity pairs onto different relational spaces. The score function of TransR is defined as

$$\mathbf{E}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h}\mathbf{M_r} + \mathbf{r} - \mathbf{t}\mathbf{M_r}\|. \tag{2}$$

By the transformation matrix $\mathbf{M_r}$, TransR can provide more diversified translations. Besides, as an extension of TransR, TransD [6] uses different transformation matrices for *head* and *tail* which have the same matrix in TransR, to implement more dynamic embedding in the KGs with lots of multi-relation triples.

Most existing triple-based Trans models only consider the triple information but ignore the semantic information in the text description [9].To make up for this shortcoming, some extra-information-based Trans models have been explored [10]–[12] to include more semantic information into the vectors of entities. Utilizing the text descriptions in Freebase [13], Description-Embedied Knowledge Representation Learning (DKRL) [9] encodes each entity description into a text-based vector by using the encoder based on Convolutional Neural Network (CNN), and then concatenates this vector and the triple-based embedding vector obtained by TransE to be the final entity representation vector. The experiments indicate that DKRL can significantly improve the performance of KG completion. Besides, a model based on the Bi-LSTM encoder, A-LSTM [14], has been proposed to embed the entity descriptions; IKRL [15] takes the image information into vector embedding to promote the accuracy on KG completion. Recently, there have also been some

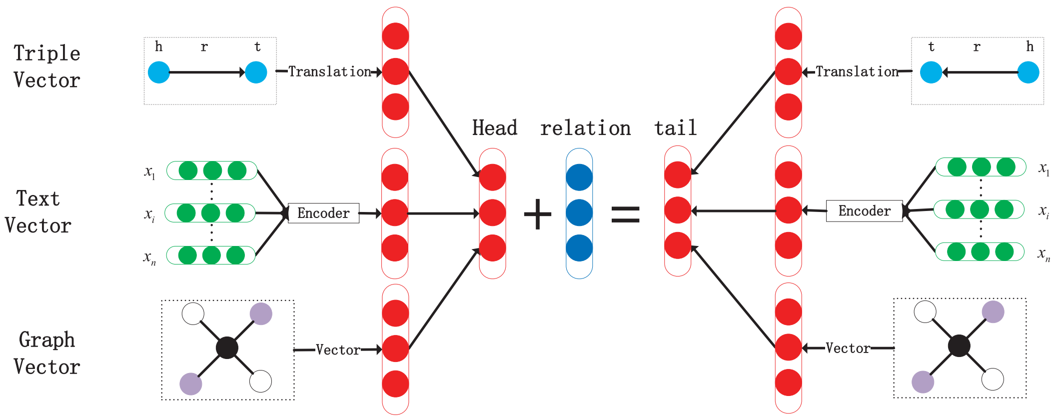


FIGURE 2. The framework of TDN model. For every entity (*head* or *tail*) in a KG, its embedding vector is composed of three sub-vectors: Triple, text and network vectors. Triple vector may still be obtained from the traditional Trans models such as TransE. Text vector, mirroring the semantic description of entity, is encoded with the text characteristics by the DKRL model. Network vector, corresponding to the topology information related with the entity, is obtained from network embedding.

approaches [16], [17] which utilize the structure information of entities to improve the embedding. But they usually only focus on the local structure information around an entity but do not handle the influences of different locations on each other, and the text descriptions are ignored.

III. TDN-BASED REPRESENTATION FRAMEWORK

TDN provides a fusion framework which also takes the text description and structure information of graph in the embedding. According to TDN, the entity embedding is defined as the following

$$\mathbf{e} = \mathbf{e_s} \oplus \mathbf{e_d} \oplus \mathbf{e_g}, \tag{3}$$

where e_s, e_d, e_g, are called triple, text, and network vector, respectively, corresponding to the embeddings of triple information, text description and network structure related with an entity, respectively; $\oplus$ is the concatenation operator.

In this paper, the embedding strictly following formula 3 is called *complete* TDN embedding. Besides, the TDN model can have some variants. When $e_d = 0$ or $e_g = 0$, $\mathbf{e} = \mathbf{e_s} \oplus \mathbf{e_g}$ and $\mathbf{e} = \mathbf{e_s} \oplus \mathbf{e_d}$ are called the *incomplete* TDN-based embedding.

Figure. 2 shows the architecture of TDN, in which e_s, e_d, e_g need to be computed as follows. Firstly, we employ the classical Trans models such as TransE [3] and TransR [5], which regard every triple as a translation from *head* to *tail* by *relation* in form of real-value vectors, to get triple vector, e_s. Secondly, the text descriptions of entity are taken into consideration. We utilize DKRL model [9], which explores continuous bag-of-words (CBOW) [18] and convolutional neural network (CNN) [19] as the text encoders. By using this approach, the semantic information of entity description is represented as a vector, e_d. Thirdly, the network structure information of KG can be obtained by the methods of network/graph embedding [20], by which each vertex in a graph (network) is represented as a low-dimensional vector, e_g, and the topology information of network is preserved. Finally, all these representation vectors of KG are integrated into a unified extended vector space in which KG is computed and analyzed with less ambiguity.

IV. METHODOLOGY

As discussed above, the TDN model includes the embeddings of triple, text and network vectors. Since we directly employ the classical Trans models to implement the embedding of triple vectors, we only introduce in detail the embeddings of text vectors and network vectors in this section.

A. EMBEDDING OF TEXT VECTORS

In this section, we separately use two models to implement the embedding of text vectors.

The first used method to represent the word embedding is *continuous bag-of-words* (CBOW) [18]. According to this method, the entity descriptions text is denoted as a word sequence $x_{i:n} = x_1, x_2, \ldots, x_n$, where x_i is the ith word in the description text. We use the average of all the word embedding vectors in the sequence as the entity description embedding:

$$\mathbf{e_d} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i}. \tag{4}$$

CBOW model can capture the key semantic information from the context [21]. Moreover, we utilize the CNN-based word embedding to build text vectors. Figure 3 shows this process. Using the same pre-processing and setting as that of DKRL, the word embedding of text descriptions is taken as the input. Two convolution layers are used in this process, and their outputs are pooled by Max-pooling and Mean-pooling, respectively. Finally, the model can produce a fixed-length representation vector for each entity without losing much information.

B. EMBEDDING OF NETWORK VECTORS

The traditional Trans model only focuses on the local information of triples, ignoring the topology information of

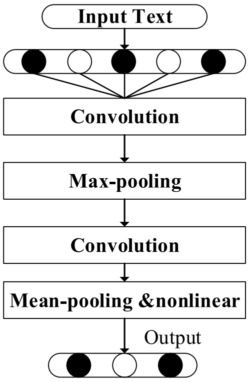


FIGURE 3. The process of CNN-based embedding.

different locations. In this study, we consider that the topology characteristics between different nodes in KG may affect each other and can provide more information for entity representation. Therefore, we use *network embedding* [20], [22] to handle this appearance and effect.

Network embedding is used to project the nodes (entities) in a graph to a low-dimensional vector space, and use the vectors to mirror the topology information of graph. According to this idea, the triples in KG can be regarded as an adjacency list denoting a network, in which the nodes and edges correspond to the entities and relations, respectively. Then, this network generated is learned by using network embedding, and the nodes are projected into the low-dimensional vector space. The foundation behind this process is that the nodes related to the similar topology information in the network should correspond to the vectors close to each other in the vector space.

DeepWalk [20], a classical network embedding model, is adopted in this study. DeepWalk generalizes recent advancements in language modeling and unsupervised feature learning from sequences of words to graphs. It uses local information obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences. Figure 4 shows the learning process of DeepWalk.

By introducing a mapping function $\Phi : \{v\} \rightarrow \mathbb{R}^{|V|\times d}$, where V and d denote the set of nodes and the dimension of vectors, respectively, DeepWalk implements the vector embedding of an arbitrary node v. Estimating the likelihood of a specific sequence of words appearing in a corpus, the problem is to estimate the likelihood:

$$Pr(v_i|(\Phi(v_1), \Phi(v_2), \ldots, \Phi(v_{i-1}))). \tag{5}$$

However, as the walk length grows, it is hard to directly compute this conditional probability. To simplify the computation, DeepWalk uses one word to predict the context instead of using the context to predict a missing word. In terms of vertex representation modeling, this yields the following optimization problem

$$\arg\min_{\Phi} \; -\log Pr(\{v_{i-w}, \ldots, v_{i+w}\} \setminus v_i|\Phi\,(v_i)), \tag{6}$$

where $\Phi(v_i)$ corresponds to the obtained network vector of node v_i.

V. TRAINING MODEL

A. NOTATIONS

Given a KG, let $T = \{(h, r, t)|h, t \in E, r \in R\}$ be the set of triples, where E is the set of entities and R is the set of relations. E and R can compose a graph $G = (E, R)$, and they correspond to the sets of nodes and edges of G, respectively. Correspondingly, the TDN model can be stated as a parameter set $\theta = \{\mathbf{X}, \mathbf{E}, \mathbf{R}, \mathbf{N}\}$ where $\mathbf{E}$, $\mathbf{R}$ stand for the triple vector embeddings of E and R, respectively; $\mathbf{X}$ stands for the text vector embedding of E; $\mathbf{N}$ stands for the network vector embedding of graph G.

B. TRAINING

According to formula (3), we combine triple, text and network vector as the final entity representation of training model. The final representation can be got by minimizing the following margin-based score function [3], [23] as objective for training:

$$\pounds = \sum_{(h,r,t)\in T} \sum_{(\hat{h},\hat{r},\hat{t})\in \hat{T}} max(\gamma + f(h, r, t) - f(\hat{h}, \hat{r}, \hat{t}), 0), \tag{7}$$

where $\gamma > 0$ is a hyper-parameter mirroring the margin between correct triples and incorrect triples; T is the set of correct triples like (h, r, t); $\hat{T}$ is the set of incorrect (corrupted) triples like $(\hat{h}, \hat{r}, \hat{t})$; $f(\cdot)$ is the score function [3] defined as

$$f(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|, \tag{8}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t}$ are the embedding vectors of h, r, t, respectively. The set of corrupted triples, $\hat{T}$, is constructed according to

$$\hat{T} = \{(\hat{h}, r, t)|\hat{h} \in E\} \cup \{(h, r, \hat{t})|\hat{t} \in E\} \cup \{(h, \hat{r}, t)|\hat{r} \in R\}, \tag{9}$$

which is composed of training triples with either the head or tail replaced by a random entity (but not both at the same time). Loss function (7) favors lower score values for training triplets than for corrupted triples, and is thus a natural implementation of the intended criterion. Note that for a given entity, its embedding vector is the same when the entity appears as the head or as the tail of a triplet.

C. PROCESS

The training of TDN first needs to be initialized from three aspects. For the embedding of triple vectors, $\mathbf{E}$ and $\mathbf{R}$ can be initialized by the translation-based methods such as TransE. For the text vectors, $\mathbf{X}$ can be initialized by the CBOW or CNN encoder that embed the entity textual descriptions. For the network vectors, $\mathbf{N}$ can be obtained from the network embedding model taking the global network of KG as input.

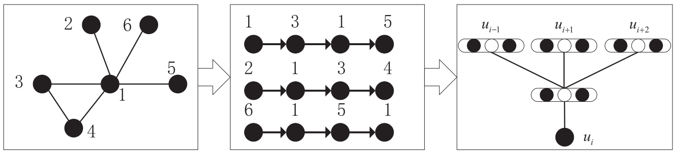


FIGURE 4. The learning process of network embedding (DeepWalk). DeepWalk implements network embedding by random walks. These walks can be regarded as some special language phrases. Starting from one node in the graph, the random walk method is used to obtain the sequence representation of node, which can be regarded as a sentence in the language model. Then the skip-gram model is trained to obtain the low-dimensional vector representation of each node.

TABLE 1. Statistics of the dataset.

Dataset	*Rel*	*Ent*	*Train*	*Vaild*	*Test*
FB15K	1,345	14,904	472,860	48,991	57,803

Then, all these initial vectors compose the initial TDN vectors according to formula (3).

After the initialization, the optimization is enforced by a standard back propagation using stochastic gradient descent (SGD). The back propagation will be blocked when meeting all-zero paddings or the current feature value was not considered in pooling during forward propagation. Finally, the TDN embeddings can be taken by minimizing the score function stated in formula (7).

VI. EXPERIMENTS

A. DATASET

In this study, we employ a dataset extracted from a real-word large-scale KG Freebase [13], called FB15K [3], to be our experimental dataset, in which the entity descriptions are publicly available in DKRL [9]. In the dataset, the average number of words in descriptions is 69 and the longest description contains 343 words. The training set has 472,860 triples and 1,341 relations; valid set has 48,991 triples, and test set has 57,803 triples.

Table 1 lists the statistics of the dataset.

B. TESTED MODELS AND PARAMETER SETTINGS

We test four groups of complete TDN-based embedding and six groups of incomplete TDN-based variants, which are explained as follows.

- TDNWE: the complete TDN embedding integrated with TransE triple vector, CBOW-based text vector and DeepWalk-based network vector;
- TDNCE: the complete model integrated with TransE triple vector, CNN-based text vector and DeepWalk-based network vector;
- TDNW(TE+CBOW): the model integrated with TransE triple vector and CBOW-based text vector, but ignoring network vector;
- TDNC(TE+CNN): the model integrated with TransE triple vector and CNN-based text vector, but ignoring network vector;
- TDN(TE+NET): the model integrated with TransE triple vector and DeepWalk-based network vector, but ignoring text vector;
- TDNWR: the complete model integrated with TransR triple vector, CBOW-based text vector and DeepWalk-based network vector;
- TDNCR: the complete model integrated with TransR triple vector, CNN-based text vector and DeepWalk-based network vector.
- TDNW(TR+CBOW): the model integrated with TransR triple vector and CBOW-based text vector, but ignoring network vector;
- TDNC(TR+CNN): the model integrated with TransR triple vector and CNN-based text vector, but ignoring network vector;
- TDN(TR+NET): the model integrated with TransR triple vector and DeepWalk-based network vector, but ignoring text vector.

We also implement two groups of current models as the baseline for comparison. The first group includes two classical Trans-based models: TransE and TransR. The second group includes DKRL+TransE [9] and A-LSTM [14], which take entity descriptions text as additional supplementary information for embedding.

We train those models with the triple vector dimension n_{tr} in $\{50, 100, 200, 300\}$, the text vector dimension n_{tx} in $\{50, 100, 200\}$ and the network vector dimension n_{nt} in $\{50, 100, 200\}$. Following most Trans-based models, we use learning rate λ in $\{0.0005, 0.001, 0.002\}$, and margin γ in $\{1.0, 2.0\}$. The parameter settings of CNN-based description Encoder are consistent with DKRL's. In our experiments, The optimal configurations of TDN are: $\lambda = 0.001, \gamma = 1.0, n_{tr} = 100, n_{tx} = 100, n_{nt} = 100$. In other words, the optimal dimension of the complete TDN embedding (integrated with triple, text and network vectors) is $n = n_{tr} + n_{tx} + n_{nt} = 300$; the optimal dimension of the incomplete TDN-based variant embedding is $n = n_{tr} + n_{tx} = 200$ or $n = n_{tr} + n_{nt} = 200$.

C. LINK PREDICTION

Link prediction is a sub-task of KG completion to complete a triplet (h, r, t) with h or t missing based on minimizing the score function. In testing phase, for each test triple (h, r, t),

TABLE 2. Results on entity prediction.

Tested Models	MeanRank		Hits@10(%)	
	Raw	Filter	Raw	Filter
TransE(200dim)	251	143	49.3	66.5
TransE(300dim)	244	141	49.6	67.8
TransR(200dim)	251	141	51.5	71.6
TransR(300dim)	253	146	51.3	71.3
DKRL+TransE	181	91	49.6	67.3
Jointly(A-LSTM)	167	77	52.9	75.5
TDNW(TE+CBOW)	154	74	54.6	75.1
TDNC(TE+CNN)	145	70	54.6	77.6
TDN(TE+NET)	143	64	54.8	**79.2**
TDNWE	142	58	54.8	76.7
TDNCE	**140**	57	55.1	**78.8**
TDNW(TR+CBOW)	147	77	54.6	74.8
TDNC(TR+CNN)	146	68	54.9	75.9
TDN(TR+NET)	147	57	54.8	77.8
TDNWR	144	**55**	**55.7**	77.9
TDNCR	**139**	**54**	**56.2**	78.3

we replace the entity by all entities in the KG, and rank these entities in descending order of similarity scores calculated by score function f defined in formula (8). Following the Trans-based models, two evaluation protocols proposed by Trans models are used as follows to evaluate the experimental results.

1) MeanRank: The average ranking of the correct entities or relationships in the triples.
2) Hits@10: The proportion of correct entities in top-10 ranked entities in the triples.

The lower MeanRank or higher Hits@10 corresponds to a better prediction. Besides, note that corrupted triple may also exist in KGs, which should also be considered as correct. This may lead to an evaluation that may underestimate those corrupted but correct triples. Therefore, we may filter out these corrupted triples before ranking. Correspondingly, we name the unfiltered setting as "Raw" and the filtered one as "Filter".

D. RESULTS AND ANALYSIS

The evaluation results are shown in Table 2, where the top two optimal results in every column are displayed in bold numbers. From the results we observe that: (1) The TDN-based embedding models significantly outperform other baseline models including TransE, TransR, DKRL+TransE and A-LSTM in both MeanRank and Hits@10, and TDNCR is the optimal model on the whole. This indicates that both text description information and network structure information can remarkably improve the accuracy of link prediction. (2) Among all the tested TDN-based embedding models, the complete TDN embedding excels over all the other models including the incomplete TDN-based embedding on the whole. This implies that not only text description information but also network structure information should be paid enough attention for KG completion. (3) The incomplete TDN-based embedding with network vector is the suboptimal on the whole, and which outperforms other models except for the complete TDN embedding. Especially, it can be seen that the Hits@10 of TDN(TE+NET) (**79.2**) is even superior to the ones of complete TDN embedding. This suggests that network structure information should play a more important role in KG completion than text description information.

VII. CONCLUSION

In this paper, we propose a TDN model for the representation learning of KG with integrated information, including triple information (T), text descriptions (D) and network structure (N). By integratedly embedding those factors in a low-dimensional vector space, TDN aims to improve the accuracy of KG completion. We give the definition of TDN-based embedding framework and explore the methodology about implementing text description embedding and network structure embedding. In experiments, we evaluate the TDN-based embedding on link prediction, and compare TDN with current models. Experimental results show that TDN model achieves better performances than other baselines on the link-prediction-based KG completion.

A method of network embedding is used to handle network structure information of KGs. By experiments, it suggests that network structure should affect link prediction more than other factors. As network embedding can be implemented by many different methods [24], [25], we need to explore more kinds of network embedding models to verify this hypothesis.

REFERENCES

[1] S. Szumlanski and F. A. Gomez "Automatically acquiring a semantic network of related concepts," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2010, pp. 19–28.
[2] H. Cai, V. M. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.
[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.
[4] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI*, Jun. 2014, pp. 1112–1119.
[5] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI*, Feb. 2015, pp. 2181–2187.
[6] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 687–696.
[7] Q. Wang, Z. Mao, and B. Wang, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.
[8] X. Dong *et al.*, "Knowledge vault: A Web-scale approach to probabilistic knowledge fusion," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 601–610.
[9] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. AAAI*, Mar. 2016, pp. 2659–2665.
[10] N. Lao, A. Subramanya, F. Pereira, and W. W. Cohen, "Reading the Web with learned syntactic-semantic inference rules," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. Assoc. Comput. Linguistics*, Jul. 2012, pp. 1017–1026.
[11] B. Taskar, V. Chatalbashev, D. Koller, and D. Koller, "Learning structured prediction models: A large margin approach," in *Proc. 22nd Int. Conf. Mach. Learn.*, Aug. 2005, pp. 896–903.
[12] A. Neelakantan, B. Roth, and A. McCallum, "Compositional vector space models for knowledge base inference," in *Proc. AAAI Spring Symp. Ser.*. Mar. 2015, pp. 31–34.
[13] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2008, pp. 1247–1250.

[14] J. Xu, K. Chen, X. Qiu, and X. Huang. (2016). "Knowledge graph representation with jointly structural and textual encoding." [Online]. Available: https://arxiv.org/abs/1611.08661

[15] R. Xie, Z. Liu, H. Luan, and M. Sun. (2016). "Image-embodied knowledge representation learning." [Online]. Available: https://arxiv.org/abs/1609.07028

[16] W. L. Hamilton, R. Ying, and J. Leskovec. (2017). "Representation learning on graphs: Methods and applications." [Online]. Available: https://arxiv.org/abs/1709.05584

[17] J. Feng, M. Huang, Y. Yang, and X. Zhu, "GAKE: Graph aware knowledge embedding," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers.*, Dec. 2016, pp. 641–651.

[18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Systems.*, 2014, pp. 3104–3112.

[19] R. Collobert *et al.*, "Natural language processing (Almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.

[20] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[21] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. (2014). "A Convolutional Neural Network for Modelling Sentences." [Online]. Available: https://arxiv.org/abs/1404.2188

[22] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.

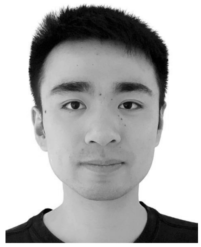
[23] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proc. Adv. Neural Inf. Process. Systems.*, 2013, pp. 926–934.

[24] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[25] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web.*, May 2015, pp. 1067–1077.

XIAOJUN KANG received the B.Sc. degree in computer science and technology from the China University of Geosciences, Wuhan, China, in 2000, and the Ph.D. degree in resource and environment information engineering from Huazhong Agricultural University, Wuhan, in 2011. He is currently an Associate Professor with the School of Computer Science, China University of Geosciences. His current research interests include artificial intelligence, natural language processing, and bioinformatics.

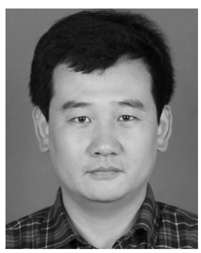
HONG YAO (M'14) received the B.Sc. degree in computer and applications from the Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 1998, and the Ph.D. degree in computer science and technology from the Huazhong University of Science and Technology, Wuhan, in 2010. He is currently a Professor with the School of Computer Science, China University of Geosciences, Wuhan. His current research interests include knowledge graph, wireless and mobile networks, and mobile cloud computing.



QINGTAO LI received the B.Sc. degree in network engineering from the China University of Geosciences, Wuhan, China, in 2016, where he is currently pursuing the master's degree with the School of Computer Science and Technology. His research interests include knowledge graph, knowledge representation, and network embedding.

XINCHUAN LI received the B.Sc. degree in resource exploration engineering and the Ph.D. degree in geographic information system from the China University of Geosciences, Wuhan, China, in 2000 and 2009, respectively, where he is currently a Lecturer with the School of Computer Science. His current research interests include informatization of geology and mining, 3D geological modeling, 3D space data management, artificial intelligence, and natural language processing.

CHAO LIU received the M.S. degree in computer application from the China University of Geosciences, Wuhan, China, in 2006, where he is currently a Lecturer with the School of Computer Science. His research interests include knowledge graph, cloud computing, and big data processing.



LIJUN DONG received the B.Sc. degree in mechatronic engineering from the Nanjing University of Science and Technology, China, in 1999, and the Ph.D. degree in computer science from the Huazhong University of Science and Technology, China, in 2008. He is currently an Associate Professor with the College of Computer Science, China University of Geosciences, China. His current research interests include knowledge graph, complex networks, and cyberspace security.