



Article ID 1007-1202(2023)01-0045-08

DOI <https://doi.org/10.1051/wujns/2023281045>

Multilingual Knowledge Graph Completion Based on Structure Features of the Dual-Branch

□ ZHANG Shijie, GAO Yongbin[†], YANG Shuqun

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201600, China

© Wuhan University 2023

Abstract: With the development of information fusion, knowledge graph completion tasks have received a lot of attention. Some studies investigate the broader underlying problems of linguistics, while embedding learning has a narrow focus. This poses significant challenges due to the heterogeneity of coarse-graining patterns. Then, to settle the whole matter, a framework for completion is designed, named Triple Encoder-Scoring Module (TEsm). The model employs an alternating two-branch structure that fuses local features into the interaction pattern of the triplet itself by perfectly combining distance and structure models. Moreover, it is mapped to a uniform shared space. Upon completion, an ensemble inference method is proposed to query multiple predictions from different graphs using a weight classifier. Experiments show that the experimental dataset used for the completion task is DBpedia, which contains five different linguistic subsets. Our extensive experimental results demonstrate that TEsm can efficiently and smoothly solve the optimal completion task, validating the performance of the proposed model.

Key words: embedding learning; structure feature; information fusion; knowledge completion

CLC number: TP 18

0 Introduction

Knowledge Graphs (KGs) such as DBpedia and Freebase encode statements about the world around us. They have attracted increasing attention from multiple fields, including question answering, knowledge inference, recommendation systems, and so on. By their very nature, KGs are far from complete as the world evolves continuously. This motivates work on automated prediction of new knowledge based on known facts. In inference tasks, KG completion has become a major focus of statistical feature learning.

KG encodes structured information about entities and their relationships as directed labels in a multilingual language. Many researchers encode entities and relations into low-dimensional vector spaces^[1]. Various efforts have attempted to address the data sparse representation challenge. Several studies have shown that reducing data variability improves the convergence of global models. However, they typically require modifying the embedding, which may result in the loss of important data about the intrinsic diversity of the local distribution. So, some methods stabilize the translation method by adjusting the distance and global model bias across the parameter space, such as TransE^[2], TransD^[3], and TransR^[4].

Received date: 2022-09-19

Foundation item: Supported by Science and Technology Innovation Action Plan" of Shanghai Science and Technology Commission for Social Development Project (21DZ1204900)

Biography: ZHANG Shijie, female, Master candidate, research direction: knowledge graph fusion and completion. E-mail: zzsjpg@163.com

[†] To whom correspondence should be addressed. E-mail: gaoyongbin@sues.edu.cn

Other studies, such as RESCAL^[5], Dist-Mult^[6], and ComplEx^[7] improve the generalization ability of the model by semantic matching bilinear strategies. In addition, some studies treat a relation as a rotation from head entity to tail entity. RotatE^[8] is proposed as the first model to handle symmetry/anti-symmetry. We admit that the optimality of parties in heterogeneous multilingualism is fundamentally at odds with the global optimality point. Then, completing the prediction is not a trivial technical challenge.

In view of the above corollary, we propose a dual-branch learning framework called Triple Encoder-Scoring Module (TEsm), which is a typical personalized joint learning frame. Specifically, we apply a convolutional network that decouples the target graph into a base encoder that participates in collaborative training and a locally preserved personalization layer. The base encoder layer learns the global feature knowledge, while the personalization layer retains the translation information to resist the embedding leakage of gradients. Each party's local training is corrected from a global perspective using global class center self-learning. A global center is defined as the average vector of each shared representation. Further, inspired by Chen^[9], TEsm dramatically reduces the computational complexity by only using iterative and alternating binary branches for training, instead of repeating sample pairs and large batches through joint learning. TEsm significantly outperforms state-of-the-art learning algorithms on multilingualism. For instance, TEsm achieves 83.3% Hits@10 (accuracy of the top 10 candidate entities) on the Japanese dataset, while the best top-1 accuracy of existing studies is 66.1%. TEsm improves the accuracy by 26% compared with most algorithms. So, the main innovation of this paper is the following:

1) We propose a new personalized learning framework to solve the distribution embedding, which mitigates the local and global problem by introducing the global dual-branch model with joint knowledge and alignment learning to correct local training.

2) We design the local layered network architecture to learn the global knowledge through self-learning effectively, and joint loss function after a classifier algorithm is used to "filter" candidate entities.

3) We implement TEsm and conduct extensive experiments on different datasets. The results show that TEsm outperforms state-of-the-art methods regarding inference accuracy and computational efficiency.

1 Related Work

1.1 Representation Learning

KG embedding is derived from the idea of word embedding. Each entity and relation is mapped to a low-dimensional vector space to measure the accuracy of the triples. For instance, the Trans model was proposed by Bordes *et al*^[2], in which translation distance used a specific relation vector r to measure the distance between two entity vectors h and t . Besides, RotatE^[8] employs complex embedding space to a model relationship r as the rotation of the vector from header entity h to tail entity t . In addition, unstructured models, such as KD-CoE^[10], AttrE^[11], MultiKE^[12], and HMAN^[13], utilize supplementary information about Wikipedia entity descriptions to improve representation learning. Later, many modifications of the above model result in several models that achieve satisfactory results with respect to data sparsity. Since there is already a large body of work in this area, we provide only a highly general summary.

1.2 Graph Neural Network

Graph Neural Network (GNN) learns node vectors by recursively aggregating the representations of neighboring nodes. GNN is applied to various processing tasks of natural languages, such as semantic role tags^[14] and machine translation^[15]. Extensive variational networks have emerged on GNN, such as Graph Convolutional Network (GCN)^[16], Relationship Graph Convolutional Network (R-GCN)^[17], and Graph Attention Network (GAN)^[18]. Because of this, the KG will be allowed to acquire more facts and hence to be better completed smoothly due to the directivity of the nodes and the structural aggregation from single points to local first-order nodes. The word vectors are obtained by the above model due to its excellent ability in modeling graph structures. So, it increases the aggregation of features to handle accurate word vector representations.

1.3 Integrated Inference

Integrated inference can achieve the best learning performance and improve the generalization ability of the algorithm by training multiple learners and combining outcomes according to a certain strategy. One of the most usual methods is Boosting^[19], followed by AdaBoost^[20] and RankBoost^[21]. AdaBoost starts from a pool of weak classifiers and iteratively selects the best one based on the sample weights in that iteration. The final classifier is a linear combination of the selected weak

classifiers, where each classifier is weighted by its performance. At each iteration, the sample weights are updated based on the selected classifier, so that subsequent classifiers focus more on hard samples. Multilingual KGs use fused source information, which then helps the target graph to predict missing facts. In this paper, we extend AdaBoost to combine ranking results from multiple KG embedding models that do an excellent job of selecting candidate entities.

2 Method

KGs are known as the data source. $KG = \{E, \mathbb{R}\}$. It is composed of several triples $\{h, r, t\}$, where h, t include the head and tail entities of triples, $\{h, t\} \in E$, and r represents the relationship between head and tail entities, and $\{r\} \in \mathbb{R}$.

2.1 Preliminary Knowledge

This paper performs optimization based on the work of multilingual completion^[9]. The framework is depicted in Fig. 1 and consists of the following three main points:

- 1) Using joint operators fuse local structural information to achieve the interaction between triples.
- 2) Dual-branch structure. Knowledge Model (KM) learns semantic vectors to encode entities and relationships, while Alignment Model (AM) uses the supervised method to obtain correspondence sets in different graphs.
- 3) Design the local layered network architecture to effectively learn the global knowledge through self-learning and joint loss function.

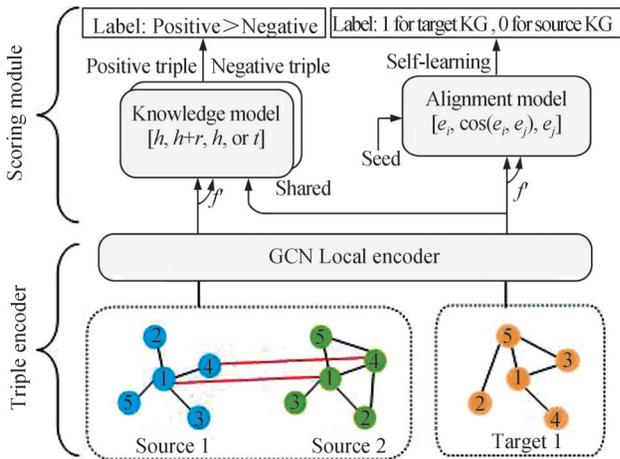


Fig. 1 Network design (TEsm)

1-5 indicate entities in the graph, and the lines between Source 1 and 2 indicate aligned seeds

2.2 Triple Encoder

Graph convolutional encoding is a vertex representation that integrates features of relations between adjacent vertices towards the spatial domain. The process requires semantic knowledge and structure data to provide a rich word vector representation for learning and alignment learning (overlapping parts are represented by sharing). The local aggregates are depicted in Fig. 2. Two methods are jointly learned:

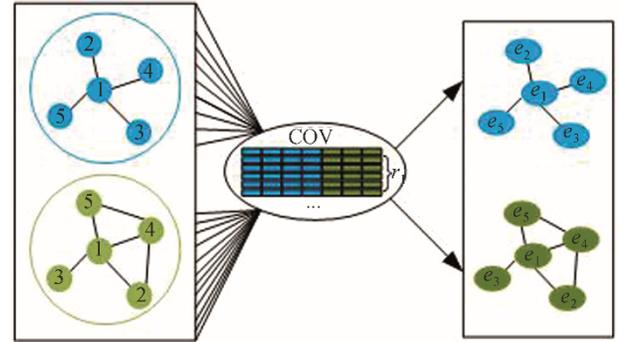


Fig. 2 Local encoder

- 1) Neighbor aggregation. GCN encodes local features to better capture entity proximity information.
- 2) Relationship model. A relation is an arithmetic operation on a vector space. The goal is to fully exploit structural and entity information proximity factors for fine-grained entity embedding.

To achieve the above, the encoder stacks the layers of the GCN network inside the KG. $E^{(l)}$ of l layer is calculated as:

$$E^{(l)} = \mathcal{O} \left(\mathbf{D}^{\frac{1}{2}} \tilde{\mathbf{A}} \mathbf{D}^{\frac{1}{2}} E^{(l-1)} \mathbf{M}^{(l-1)} \right) \quad (1)$$

where \mathbf{D} represents the diagonal matrix, which is the input feature of the node. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the sum of the adjacency matrix \mathbf{A} and identity matrix \mathbf{I} . $\mathbf{M}^{(l-1)}$ is $l-1$ layer weight matrix. The result of the convolution operation performed by the activation function \mathcal{O} is passed to the network layer. The initial features $E^{(0)}$ of an entity can be properties of the entity itself or randomly generated. $E^{(l-1)}$ is the output result of the previous layer and the last layer $E = E^{(l)}$ represents an entity embedding.

2.3 Knowledge Model (KM)

L_K^G represents the score function (2) of language m in the knowledge graph G :

$$L_K^G = \sum_{h_E, r, t_E \in G} \left[\max(0, -f(h_E, r, t_E) + f(h'_E, r, t'_E) + y) \right] \quad (2)$$

where $[\cdot] = \max(0, \cdot)$, f is a specific triplet score function,

which is used in the following functions in ① and ② below. γ is a hyperparameter, (h_E, r, t_E) is a positive sample, G is a set of actual triples in the KG. A negative sample (h'_E, r, t'_E) generated by randomly replacing the true triplet with either a head entity or a tail entity, and G' is the set of constituent negative samples. Notably, implementing this procedure requires finding positive and negative samples in a uniform vector space, and we mainly consider two representative triplet scoring functions.

① TransE-GCN TransE translates relations into translations between head and tail entities in Euclidean space. The formula is as follows:

$$f_{\text{TransE}}(h_E, r, t_E) = -(h_E + r - t_E)_2 \quad (3)$$

where (h_E, r, t_E) is the embedding vector graph convolution l in the last layer.

② RotatE-GCN RotatE treats relations as rotations in a complex space, and tail entities are derived from head entities that rotate through relations in the vector space. The formula is as follows:

$$f_{\text{RotatE}}(h_E, r, t_E) = -(h_E \circ r - t_E)_2 \quad (4)$$

where \circ represents the Hadama product in complex vector space.

2.4 Alignment Model (AM)

Alignment information between each pair of embeddings is connected by additional property labels of the graph structure, and entity alignment is achieved using self-supervised learning. Given a set of different language graphs G_i and G_j , the entity correspondence between any two graphs requires matching based on a small number of $e_i \in G_i, e_j \in G_j, |G_i| \leq |G_j|$ seed entities. Alignment function (5) is defined using cosine similarity:

$$L_A^{G_i \leftrightarrow G_j} = \sum_{(e_i, e_j) \in \Gamma_{G_i \leftrightarrow G_j}} (\cos(e_i, e_j))_2 \quad (5)$$

where $G_i \leftrightarrow G_j$ represented the two graphs as seed data $(e_i, e_j) \in \Gamma_{G_i \leftrightarrow G_j}$. In particular, Nearest Neighbor (NN)^[22] adaptive learning of the relationship is used during alignment to reduce the candidate after each iteration.

2.5 Joint Model

The above model is a two-branch structure that optimizes the loss per function for entities and relations, and multiple KGs are trained uniformly and jointly. The formula is as follows:

$$\Gamma = \sum_{m=1}^M L_{K^m}^{G_m} + \lambda \sum_{i=1}^M \sum_{j=i+1}^M L_A^{G_i \leftrightarrow G_j} \quad (6)$$

where $L_{K^m}^{G_m}$ is the loss of the knowledge model as defined in Eq. (2), $L_A^{G_i \leftrightarrow G_j}$ is alignment loss in Eq. (5), and λ is the weight of the two models that forms the positive value hyperparameter. There is no direct optimization function Γ . Instead, each iterates through different batches and optimizes each function loss separately. The value of M is 5, representing the other five datasets in the paper. In addition, L2 regularization is used to prevent overfitting problems.

It is shown that only very high initial cosine similarity can achieve an equivalent computation that maintains stable alignment within a few iterations. In this way, the above learning patterns can be efficiently constructed using an iterative process with equal batches of joint training.

2.6 Integrated Reasoning

Knowledge transfer and fact completion are based on the results of the above model. The target query is transformed into the source KG, and the result is then transformed back to the target graph by aligning the information. It is weighted and summed using an ensemble of classifier algorithms.

As shown in Fig. 3, given an incomplete triplet, the completion goal is to accurately find the tail entity, which forms a new triplet in the target graph. A weighted evaluation scheme is used to systematically rank and validate candidate value entities.

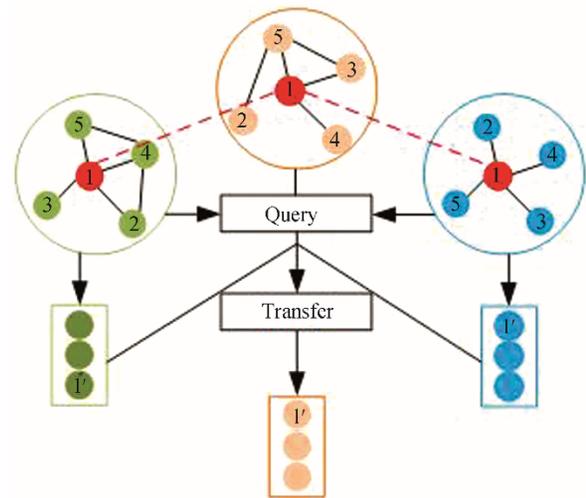


Fig. 3 Integrated reasoning

A triple group $(h, r, ?)$ query the corresponding two source map entities and deduce the ranking of their triples; Finally, the final target candidate entity is determined by the transfer

First, the source and target aligned entity pairs are predicted, and then the source is queried to the target based on the alignment results. The information from sources 1 and 2 is used to complete the target graph, and the result obtains the candidate results by the target graph. The final candidate entities are predicted by weighting^[21]. The formula is as follows:

$$s(e) = \sum_{i=1}^m w_i(e) N_i(e) \quad (7)$$

where e represents an entity in the target graph and $w_i(e)$ represents a specific entity model weight. The AdaBoost classifier is used in the computation to learn each specific weight. If an entity e is in top- k (where k is selected as 1,3,10), $N_i(e)$ is 1 or 0.

3 Experiment and Result Analysis

3.1 Experimental Setting

This section evaluates and validates the importance of each module for TEsm. A detailed study demonstrates the effectiveness and generalization of the model.

3.1.1 Experimental dataset

All experiments in this paper are based on the DBP-5L dataset. Moreover, the multilingual alignment task is relatively late-stage, with only a few suitable datasets providing seed alignment between entity and relation pairs. In this way, the matching dataset is dominated by the five languages derived from DBpedia, namely English (En), Greek (El), Spanish (Es), Japanese (Ja), and French (Fr), as listed in Table 1. In the DBP5L dataset, about 40% of entities in one language are aligned with entities in another language. The relations of the five graphs are represented in a uniform pattern, consistent with the definition of the problem.

Table 1 Statistics of DBP-5L dataset

Item	Multilingual academic KG (DBP-5L)				
	En	Ja	El	Fr	Es
Entity	13 996	11 805	5 231	13 176	12 382
Relation	831	128	111	178	144
Triple	80 167	28 774	13 839	49 015	54 066

All language graphs are aligned to collect triples gradually. The entity alignment covers approximately 40% of the distance between any two graphs. Based on the same set of seed entities, the El KG has significantly

less vocabulary than the other four. The dataset is divided into training, validation, and testing at a ratio of 60:30:10. Relational labels represent only 8% of actual triplets, meanwhile, almost 80% of genuine triplets express relations across all languages. The study also randomly selects half of the seed entities to be aligned for training and evaluation.

3.1.2 Evaluation method

All other entities in the graph are wrong "candidates" to replace their head or tail entities. After the previous work, the "Filter" setting is used, and the candidate space for the post triple has been excluded. The goal of the model is to select the correct triple with the "filter" method. Each test set is considered as a query and the top- k predictions are retrieved. In this paper, the Hits@ k metric refers to the average proportion of triples ranked less than n in link prediction. In general, take n to be equal to 1, 3, or 10. When n is 10, the ranking of candidate entities is more accurate.

3.1.3 Experimental environment

The equipment selected for the training part of this experiment is as follows: a server running Ubuntu18.04 and using a graphics card NVIDIA RTX 8000. The algorithm is implemented using Python3.6 and the TensorFlow 1.10 framework. In the training process, Adam^[22] is used as an optimizer, and the hyperparameter fine-tunes Hits@ k through a grid search. The learning rate $lr \in \{0.1, 0.01, 0.001, 0.0001\}$, the dimension $d \in \{32, 64, 128, 200, 300\}$, batch size $b \in \{64, 256, 512, 1024\}$. This parameter is determined by the accuracy of the validation set, and the L2 regularization coefficient is fixed at 0.0001. The boundary is 0.5. TEsm (TransE) is best set to $\{lr = 0.001, d = 300, b = 256\}$. TEsm (RotatE) is best set to $\{lr = 0.01, d = 200, b = 512\}$.

3.2 Accuracy Results

3.2.1 Main results

The results for datasets are reported in Table 2. Effects on the baseline models TransE, RotatE, DistMult, TransD and HolE are taken from Ref. [8]. RotatE-booting and TransE-booting are obtained from Ref. [9]. GCN combines the methods of TEsm model and ensemble inference to improve Hits@10 by 1.1% to 13.0%. As shown in Table 2, the Ja dataset is used as the target grape and the other four are used as source grapes for completion. Compared with the baseline data, RotatE-GCN is improved by 48%, Hits@3 by 15%, and Hits@1 by 0.4%. The En dataset is used as the target graph and the other four are used as source grapes for completion.

Table 2 Completion result of different language %

Language	Method	Hits@10	Hits@3	Hits@1
Ja	TransD	33.1	20.8	1.2
	DisMult	30.7	18.6	10.4
	HolE	29.9	32.4	27.1
	RotatE	35.7	19.0	21.1
	TransE	48.5	34.4	21.1
	RotatE-booting	56.4	39.8	26.7
	TransE-booting	66.1	39.5	26.7
	TransE-GCN(TEsm)	58.1	44.5	25.3
	RotatE-GCN(TEsm)	83.3	45.3	27.8
En	TransD	29.6	17.6	3.3
	DisMult	29.4	19.9	9.1
	HolE	25.5	21.7	12.5
	RotatE	29.7	26.7	12.4
	TransE	29.3	16.4	7.1
	RotatE-booting	30.3	25.9	12.3
	TransE-booting	32.0	21.8	11.7
	TransE-GCN(TEsm)	38.3	24.1	11.1
	RotatE-GCN(TEsm)	34.7	22.1	13.4
Es	TransD	14.1	22.4	2.7
	DisMult	22.9	14.8	8.1
	HolE	30.2	27.7	18.3
	RotatE	53.9	31.1	21.2
	TransE	45.0	29.3	13.2
	RotatE-booting	60.6	40.4	21.7
	TransE-booting	46.3	32.1	17.2
	TransE-GCN(TEsm)	52.9	32.9	15.7
	RotatE-GCN(TEsm)	60.6	40.7	23.3
Fr	TransD	26.4	13.2	5.1
	DisMult	25.3	14.1	6.9
	HolE	28.1	25.5	19.0
	RotatE	55.5	29.4	23.2
	TransE	48.8	33.1	17.5
	RotatE-booting	60.8	32.1	25.9
	TransE-booting	53.3	35.7	19.0
	TransE-GCN(TEsm)	56.8	36.8	19.5
	RotatE-GCN(TEsm)	66.4	26.6	25.3
El	TransD	22.4	12.1	3.2
	DisMult	11.7	13.1	8.3
	HolE	18.3	11.2	4.8
	RotatE	36.2	18.8	14.7
	TransE	40.6	21.1	12.6
	RotatE-booting	51.2	44.3	27.5
	TransE-booting	66.7	41.9	23.4
	TransE-GCN(TEsm)	67.1	42.1	23.8
	RotatE-GCN(TEsm)	52.5	42.7	23.8

Compared with the baseline data, the experimental results are improved by 26% for TransE-GCN, 30% for Hits@1, and 14% for RotatE-GCN with Hits@1. The Es dataset is used as the target graph and the other four are used as source grapes for completion. RotatE-GCN improves the baseline by 26% on Hits@10, 10% on Hits@3 and 35% on Hits@1. The Fr dataset is used as the target graph and the other four are used as source grapes for completion. RotatE-GCN improves Hits@10 by 24% and Hits@3 by 14% over baseline TransE. The El dataset is used as the target graph and the other four are used as source grapes for completion. Compared to baseline data, TransE-GCN improves Hits@10 by 31% over baseline TransE. In the face of sparse data, the performance of the simple model is not greatly degraded compared with the accuracy of the full data. Does this mean that the performance of the simple model is sufficient for some simple entities, and that the complex model actually overfits the data? As can be observed, TEsm performs best on datasets with varying levels of heterogeneity. The model presented in this paper outperforms the base model. Concerning sparse semantic interference, it is also observed that TEsm improves in sparse graphs compared with dense ones. In particular, the multilingual (Greek) KG with a small number of entities Hits@1 increased by 13% over the baseline. This finding confirms that low knowledge coverage and sparse structure favor additional knowledge.

It is worth elaborating on the comparison between our model and the bootstrap model that inspired our work. The experimental results showed that our models (TransE-GCN, RotatE-GCN) both consistently yielded better results with improvements of 10.8% and 6.7% in terms of multilingual five datasets, respectively. We believe that the improvement is attributable to two reasons. First, due to the idea of heterogeneous features of KGs proposed, semantic embedding successfully captures both local structural information by considering entities and relations in the neighborhood, as well as the semantic information residing within the transformation operator. Moreover, by doing so, the dual-branch model of relations in a KG is modeled only once. Simultaneous replacement of previously entity-specific matrices by shared ones could potentially facilitate encoding of more complex latent information. As a result, fewer parameters need to be learned in our model, which helps alleviate the overfitting problem. To explore the contribution of each module, the following ablation experiments are

performed. KM represents the knowledge learning model in Section 2.3 and AM represents the alignment model in Section 2.4. Extensive ablation experiments have been performed with this model. The specific ablation methods are as follows:

1) For the entire TEsm framework, remove the components from Eq. (1) to estimate the importance of structure information.

2) Picking out the GCN encoder for different model

acquire triplet embedding. Depending on the combination, only suboptimal results can be obtained.

3) Accuracy is measured by differential TransE or RotatE combined with Eq. (2).

The results are presented in Table 3. The accuracy of the proposed model differs from that of the base translation model. The performance of either the knowledge model or the alignment model is improved compared with the baseline.

Table 3 GCN for RotatE and TransE model generalization performance

Model	Group	KM	AM	TEsm	Hits@ <i>k</i>	Japanese	English	French	Spanish	Greek
RotatE	1	√		√	10	63.4	31.7	59.4	55.2	51.3
					3	31.6	25.8	28.8	39.6	40.6
					1	20.0	12.5	22.3	19.7	25.0
	2			√	10	66.1	30.8	59.3	51.7	50.6
					3	30.7	27.0	28.9	38.4	38.4
					1	24.4	13.1	22.6	21.3	25.2
	3	√	√	√	10	83.3	34.7	66.4	63.0	52.5
					3	45.3	22.0	26.6	38.3	42.6
					1	27.8	13.4	16.3	23.6	27.8
TransE	1	√		√	10	55.7	29.4	49.1	46.7	44.4
					3	36.1	17.1	34.6	32.0	21.7
					1	22.4	8.9	17.1	12.8	13.1
	2			√	10	54.9	29.1	49.1	47.1	50.1
					3	37.5	18.1	33.0	30.8	21.3
					1	22.5	7.7	17.9	13.2	13.0
	3	√	√	√	10	64.5	38.3	56.8	52.9	62.1
					3	44.5	24.1	36.8	32.9	46.1
					1	25.7	11.1	19.5	15.7	23.8

√ denotes that the TEsm acts on KM and AM respectively, the parameter *k* of Hits@*k* is set to 1, 3, 10

Above, both models better learn the internal information of the triplet during the fusion of local knowledge. The accuracy of the full multilingual Hit@10 is improved by 30%. Thus, GCN(TEsm) generalization and robustness of the model are verified.

4 Conclusion

In this paper, we propose a unified framework for simultaneous knowledge acquisition and alignment. To handle the characteristic features of KGs when using traditional translation models, we develop a novel approach to transform a neighborhood into a homogeneous neighborhood via GCN. The embedding space captures both structured and unstructured vectors based on local

features. In addition, the AdaBoost classifier selects an auxiliary input of the source graph to aid the goal completion. Experimental results demonstrate the superiority of the TEsm model, showing the effectiveness of handling completion.

In the future, besides graph structure, there are other multimodal improvement learning methods including text, speech, pictures, and videos. Realizing cross-modal knowledge will be quite challenging.

References

- [1] Dettmers T, Minervini P, Stenetorp P, *et al.* Convolutional 2D knowledge graph embeddings [J]. *Proceedings of the*

- AAAI Conference on Artificial Intelligence*, 2018, **32**(1): 1811-1818.
- [2] Bordes A, Usunier N, Duran A G, *et al.* Translating embeddings for modeling multi-relational data [J]. *Advances in neural Information Processing Systems*, 2013, **26**(1282): 2787-2795.
- [3] Ji G L, He S Z, Xu L H, *et al.* Knowledge graph embedding via dynamic mapping matrix [C]//*Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Stroudsburg: Association for Computational Linguistics, 2015: 687-696.
- [4] Lin Y K, Liu Z Y, Luan H B, *et al.* Modeling relation paths for representation learning of knowledge bases [EB/OL]. [2022-08-05]. <https://arxiv.org/abs/1506.00379>.
- [5] Nickel M, Tresp V, Krieger H P. A three-way model for collective learning on multi-relational data [C]// *Proceedings of the 28th International Conference on Machine Learning*. New York: ACM, 2011: 809-816.
- [6] Yang B S, Yih W T, He X D, *et al.* Embedding entities and relations for learning and inference in knowledge bases [EB/OL]. [2022-08-29]. <https://arxiv.org/pdf/14126575.pdf>.
- [7] Trouillon T, Gaussier É, Bouchard G. Complex embeddings for simple link prediction [C]// *International Conference on Machine Learning*. New York: ACM, 2016: 2071-2080.
- [8] Sun Z Q, Deng Z H, Nie J Y, *et al.* RotatE: Knowledge graph embedding by relational rotation in complex space [EB/OL]. [2022-08-29]. <https://arxiv.org/abs/1902.10197>.
- [9] Chen X. Multilingual knowledge graph completion via ensemble knowledge transfer [EB/OL]. [2022-06-08]. <https://arxiv.org/pdf/201003158.pdf>.
- [10] Trisedya B D, Qi J Z, Zhang R. Entity alignment between knowledge graphs using attribute embeddings [J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, **33**(1): 297-304.
- [11] Zhang Q H, Sun Z Q, Qu Y Z. Multi-view knowledge graph embedding for entity alignment [EB/OL]. [2022-08-06]. <https://arxiv.org/pdf/190602390.pdf>.
- [12] Yang H W, Zou Y Y, Shi P, *et al.* Aligning cross-lingual entities with multi-aspect information [EB/OL]. [2022-08-20]. <https://arxiv.org/abs/1910.06575>.
- [13] Marcheggiani D, Titov I. Encoding sentences with graph convolutional networks for semantic role labeling [EB/OL]. [2022-07-30]. <https://arxiv.org/pdf/170304826.pdf>.
- [14] Bastings J, Titov I, Aziz W, *et al.* Graph convolutional encoders for syntax aware neural machine translation [EB/OL]. [2022-08-18]. <https://arxiv.org/pdf/170404675.pdf>.
- [15] Thomas N, Kipf T N, Welling M. Semi supervised classification with graph convolutional networks [EB/OL]. [2022-08-22]. <https://arxiv.org/pdf/1609.02907.pdf>.
- [16] Schlichtkrull M, Kipf T N, Bloem P, *et al.* Modeling relational data with graph convolutional networks [C]// *European Semantic Web Conference*. Cham: Springer International Publishing, 2018: 593-607.
- [17] Veličković P, Cucurull G, Casanova A, *et al.* Graph attention networks [EB/OL]. [2022-08-26]. <https://arxiv.org/abs/1710.10903>.
- [18] Conneau A, Lample G, J'egou H. Word translation without parallel data [EB/OL]. [2022-08-30]. <https://arxiv.org/pdf/171004087.pdf>.
- [19] Kingma D P, Ba J. Adam: A method for stochastic optimization [EB/OL]. [2022-07-30]. <https://arxiv.org/pdf/1412.6980.pdf>.
- [20] Freund Y, Schapire R E. A decision-theoretic generalization of online learning and an application to boosting [J]. *Journal of Computer and System Sciences*, 1997, **55**(1): 119-139.
- [21] Freund Y, Iyer R D, Schapire R E, *et al.* An efficient boosting algorithm for combining preferences [J]. *Journal of Machine Learning Research*, 2003, **4**: 933-969.
- [22] Artetxe M, Labaka G, Agirre E. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings [EB/OL]. [2022-08-26]. <https://arxiv.org/abs/1805.06297>.

□