Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 4, pp. 1173–1184
DOI : 10.47974/JDMSC-1972

Missing relation prediction in knowledge graph using local and neighbour aware entity embedding

Anish Khobragade *
Sanket Patil †
Harsha Rathi §
Department of Computer Science and Engineering
COEP Technological University
Pune 411005
Maharashtra
India

Shashikant Ghumbre ‡
Department of Computer Engineering
Government College of Engineering and Research, Avasari Khurd
Pune 412405
Maharashtra
India

Abstract

Predicting relations involves deducing the absent connections between entities within a knowledge graph. Knowledge graph embedding has recently emerged as a technique to represent entities and their relationships in a condensed vector space. While effective for entity prediction tasks, this approach escalates in complexity, particularly for extensive knowledge graphs. This study introduces a new method for relation prediction in knowledge graphs, employing a mechanism termed Local and Neighbor Aware Entity Embedding (LNAEE). LNAEE utilizes skip-gram initialization to capture local entity features at the triple level. It then employs an attention mechanism to update the central entity's features based on its immediate neighbors. Finally, a multilayer neural network predicts the most likely relationship type between the given entity pair. LNAEE is streamlined and less intricate owing to the reduced relation search space. Benchmark dataset experiments demonstrate

* *E-mail:* anishraj.comp@coeptech.ac.in (Corresponding Author)
† *E-mail:* patilsr18.comp@coep.ac.in
§ *E-mail:* rathihs18.comp@coep.ac.in
‡ *E-mail:* shashi.ghumbre@gmail.com

© Taru Publications

that LNAEE outperforms the baselines. This makes LNAEE ideal for applications requiring a lightweight and affordable model for knowledge graph enrichment.

Subject Classification: *Primary 68T35, Secondary 68T05.*

Keywords: *Knowledge graph, Representation learning, Embedding, Relation prediction, Attention network.*

1. Introduction

The Knowledge Graph (KG) acts like a giant encyclopedia for machines. It stores information in a format the computer can understand, called Resource Description Framework (RDF). This allows KGs to be used for tasks like searching, organizing, and retrieving real-world information. Imagine the KG as a massive web of connections. Facts are stored as triplets, similar to sentences with a subject, verb, and object. Figure 1 shows the example, "Elon Musk" (subject) "was born in" (verb/relation) "Pretoria" (object). These connections help machines understand the relationships between things in the real world. There are large, publicly available KGs like Yago, WikiData, and FreeBase. However, KGs are never truly complete [1]. However, KGs may be incomplete due to omitted entities or relationships, outdated data, or partial supporting information. This inherent incompleteness can yield unreliable query results in diverse applications such as education [2], healthcare [3], and cybersecurity [4]. Bridging this gap through missing relation prediction is a pressing research area in artificial intelligence.

A recent trend in AI is Knowledge Graph Representation Learning (KGRL). This technique tackles the incompleteness of KGs by mapping entities and relations into a numerical space (embeddings). These embeddings capture the relationships between things. The model can predict missing facts within the KG by analyzing these embeddings. During training, the model tries to distinguish between real facts (positive triplets) and made-up facts (negative triplets). First, negative triplets are created by replacing the head or tail entity in a real fact with a random entity. However, using random entities can lead to gradient vanishing problems, making it harder for the model to learn effectively. Second, when predicting a single missing fact, the model must consider all possible replacements for the missing entity, which can be very computationally expensive, especially for large knowledge graphs.

The objective of this study is to efficiently predict the absent connection (relation) between two existing entities (head-tail) within a

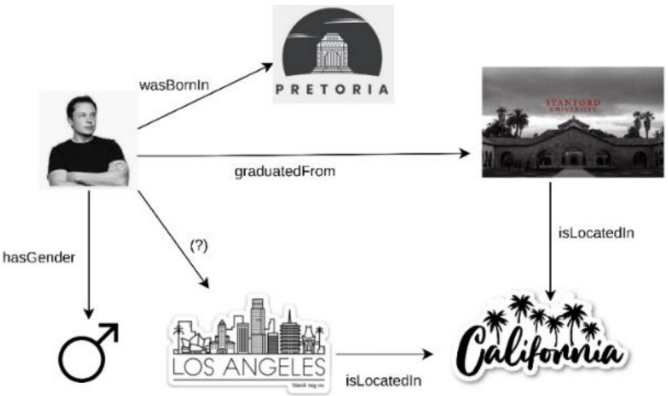


Figure 1

KG subgraph from the YAGO3-10

knowledge graph. As illustrated in Figure 1 using data from the YAGO3-10, imagine a scenario where a user queries the relation between "ElonMusk" and "LosAngeles," a relationship currently missing from the KG. Our approach aims to predict this missing link, which, in this case, would be "*livesIn*," making the KG more complete. Compared to predicting the missing entity (tail) for a known relation and head, predicting the missing relation type is computationally faster and easier.

There are far fewer relation types compared to the vast number of entities in the real world. Some entities have many connections to others, meaning there's a lot of valuable information to be found around them. Previous research has focused on how close entities are, either individually or within small groups. Influence from Graph Attention Network (GAT) [5] has been integrated into various research domains within the graph structure field. This paper introduces a new model called **L**ocal and **N**eighbor **A**ware **E**ntity **E**mbedding (LNAEE) for relation prediction in KG. Unlike other methods, LNAEE doesn't need to create negative examples for training. LNAEE works in three steps, all those steps are described in section 4. Here are the key takeaways of our approach:

(1) LNAEE captures local entity features using skip-gram and incorporates features of neighboring entities through an attention mechanism.

(2) The MNN is designed to estimate the likelihood (score) of all possible relations between the two entities in a triplet.

(3) Extensive experiments on standard KG completion datasets demonstrate that LNAEE outperforms existing methods.

2. Related Work

Classical TransE [6] represents relations embedding ($\boldsymbol{r}$) as translations between entities ($\boldsymbol{h}$, $\boldsymbol{t}$) embedding vector $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$ serves as the triplet score but encounters challenges with complex relation types. TransH [7] and TransR [8] offer a flexible solution to handle complex relations by projecting entity vectors onto relation-specific hyperplanes. TransD [9] further refines this concept by separating the projection matrices for triplet entities. Several models experimented with training approaches. DTransE [10] employs distributed training for TransE, utilizing a triplet partitioning algorithm. Yet another approach captures the latent semantics of KG in a partly examined tensor, learning the basic structure through tensor factorization as $\boldsymbol{h}\boldsymbol{R}_{\mathrm{k}}\boldsymbol{t}^{\mathrm{T}}$. DistMult [11] employs the diagonal of $\boldsymbol{R}_{\mathrm{k}}$ in its scoring function to capture their interactions. ComplEx [12] works with complex-valued feature vectors $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \in \mathbb{C}^k$. HolE [13] introduces the holographic associative memory using the circular correlation between entity pair $\boldsymbol{r}^{\mathrm{T}}\boldsymbol{h} * \boldsymbol{t}$. The convolution-based model ConvE [14] uses 2D convolution over concatenated head vectors and relation vectors to generate a semantic feature map combined with a tail embedding vector to score given triplets. ConvKB [15] extends this by concatenating entity pair and relation embedding vectors for 2D filters, enhancing learning efficiency. ConvR [16] utilizes relation-specific 2D filters over the head vector reshaped to a matrix by transformation operations, capturing relation-specific feature maps. ConPoly [17] uses a convolution mechanism to capture the relation polysemy present in the complex relation type.

3. Methodology

3.1 *Preliminary*

Let KG = {E, R, T}, where E signifies a set of entities, R signifies a set of relation types, and T signifies a set of existing triplets. Each triplet (h-r-t) comprises a head and tail entity, where $h, t \in E$, and relation type $r \in R$ between them. Embedding feature vectors $\boldsymbol{h}, \boldsymbol{t}, \boldsymbol{r} \in \mathbb{R}^k$, where k is the vector size. Subgraphs are centered around an entity h_i and include its 1-hop neighbors $(h_i) = \{r_l, t_j \mid (h_i, r_l, t_j) \in T\}$.

3.2 *Problem Formulation*

The LNAEE intends to predict relation type using probability distribution of relation types $r_l \in \{r_1, r_2, \ldots, r_{|R|}\}$ for a given entity pair sample $s_n = p(r_l | h_i, t_j)$ where $i \neq j$ and $h_i, t_j \in E$. The LNAEE predicts the missing relation type r_{pred} with the highest probability for the given (h, t), framing it as an optimization problem to maximize the likelihood of the semantically appropriate relation type.

$$r_{pred} = \max(p(r_1 | h,t), p(r_2 | h,t), \ldots, p(r_{|R|} | h,t)) \quad (1)$$

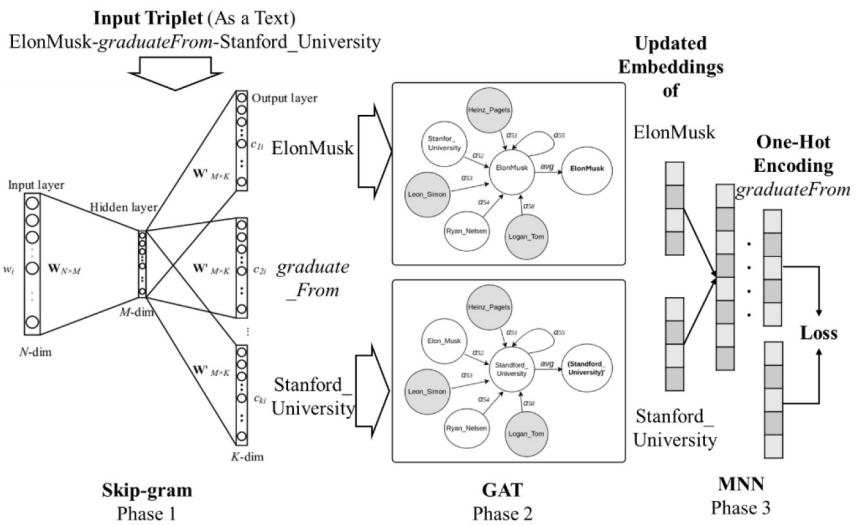


Figure 2

Proposed System Desing consists of Skip-gram to capture triplet level local proximity, GAT to update the entity features using graph structure, and MNN to compute the loss

3.3 *Local Level Semantic Proximity*

The relations and entity feature vectors are initialized to $\mathbb{R}^k$ size using the Xavier method and given as input to the word2vec (Tomas, Sutskever, Chen, Corrado, & Dean, 2013) based skip-gram approach. The set of triplet T as corpus input to the word2vec model, as shown in the first half of Figure 2, generates the feature vectors for each $\boldsymbol{h}_i, \boldsymbol{t}_j \in E$, and $\boldsymbol{r}_l \in R$. The intuition behind the utilization of word2vec is that relations occur in the context of their respective entity pairs. Equation 2 gives the context capture function as the total log probability of observing the words given

their contexts, where *Context*(*i*, *c*) represents the set of neighbors of entity '*i*' within a window of size *c*. The initialized feature vectors of entities are stacked as $\boldsymbol{X}^{|E|\times k}$ and given to the next attention layer.

$$L = \sum_{i} \sum_{j \in context(i,\ c)} \log p(e_i \mid e_j) \tag{2}$$

3.4 *Neighborhood Level Semantic Proximity*

LNAEE incorporates the GAT to assign different weights to the features of surrounding entities based on their significance to the central entity, shown in the second half of Figure 2. The attention mechanism takes $\boldsymbol{X}^{|E|\times k}$ as an input, and the outcome is a new updated embedding feature matrix $\boldsymbol{X}'^{|E|\times k'}$ as $x'_i \in \{x'_1, x'_2, \ldots, x'_{|E|}\}$ and the dimension of each entity feature vector is $\mathbb{R}^{k'}$. To obtain the updated attention-aware feature vector x'_i, the expressive power of triplet with x_i as head with tail x_j utilized to learn the attention mechanism by performing a linear transformation on concatenated vectors of x_i, x_j, being as attention coefficient. $a^T[\Theta x_i \,\|\, \Theta x_j]$. The final attention-aware feature vector of entity x'_i is obtained using Equation 3, where SoftMax σ is to normalize the attention.

$$x'_i = \sigma\Big(\sum_{j \in \mathcal{N}_{(i)}} a_{ij}\theta(x_i x_j)\Big) \tag{3}$$

where $\theta_1 \in \mathbb{R}^{k'\times 3k}$ is a shared learnable parameter weight matrix used as linear transformation assigned to an element of triplet, $a \in \mathbb{R}^{3k'}$ is a learnable weight vector used in the attention mechanism of single-layer feedforward neural network. KG substructure is captured by computing $\alpha_{i,j}$ for each head entity x_i with all associated incoming tails x_j link as $j \in \mathcal{N}_{(i)}$, where $\mathcal{N}_{(i)}$ represents all the neighborhoods. The fully expanded attention mechanism of the extended GAT model is given in Equation 4.

$$\alpha_{ij} = \frac{\exp(Leaky\,\mathrm{Re}LU\ (a^T[\Theta x_i \| \Theta x_j]))}{\sum_{j \in \mathcal{N}_{(i)}} \exp(Leaky\,\mathrm{Re}LU\ (a^T[\Theta x_i \| \Theta x_j]))} \tag{4}$$

where ‖ denotes the concatenation operation between the transformed vector of the triplet.

3.4 *Relation Prediction using MNN*

To calculate the probability distribution of relation $r_l \in \{r_1, r_2, \ldots, r_{|R|}\}$ for the provided head and tail entity (h_i, t_j). First, we create the one-hot-encoding of $r_l \in R$ of size $n = |R|$. Concatenate the entity feature vector of $\boldsymbol{h}_i, \boldsymbol{t}_j \in \{x'_1, x'_2, \ldots, x'_{|E|}\}$, as $[\boldsymbol{h}_i \,||\, \boldsymbol{t}_j] \in \mathbb{R}^{2k'}$ and pass through the MNN with

five hidden layers as shown in Figure 3. The size of the first hidden layer is $2k' - step$, although the next hidden layer size is defined by Equation 5 to the output layer of dimension $n = |R|$, where the *step* is defined by using Equation 6 to get the next hidden layer size.

$$size(HiddenLayer_{i+1}) = size(HiddenLayer_i) - step \tag{5}$$

$$step = \frac{2k' - |R|}{|HiddenLayers + 1|} \tag{6}$$

We applied the categorical cross-entropy loss as the learning objective denoted in Equation 7. Where y_i is one-hot-encoding of relation $\boldsymbol{r}_l \in R$, and $\hat{y}_i$ is the predicted output of the scoring model.

$$L = -\frac{1}{|R|}(\Sigma_{i=1}^{|R|} y_i \cdot \log(\hat{y}_i)) \tag{7}$$

4. Experimental Setup

4.1 *Datasets*

To evaluate our model's effectiveness in relation prediction, we tested it on two widely used benchmark datasets WN18-RR and FB15k-237 (Tim, Minervini, Stenetorp, & Riedel, 2018). WN18-RR focuses on capturing relationships between words in the English language, while FB15k-237 deals with real-world entities and their connections. It includes information about famous people, movies, sports, and teams. Table 1 summarizes each dataset, including how it's divided for training and testing purposes.

Table 1

Statistics of Datasets

Dataset	$\|E\|$	$\|R\|$	$\|T\|$	**#Train**	**#Valid**	**#Test**
WN18RR	40943	11	93003	86835	3034	3134
FB15k-237	14951	237	310116	272115	17535	20466

4.2 *Baselines*

We have taken six widely used KGE baselines TransE (Bordes, Usunier, Garcia-Duran, Weston, & Yakhnenko, 2013) and TransH (Zhen, Zhang, Feng, & Chen, 2014) from the translational approach, DistMult (Bishan, Yih, He, & Gao, 2015) and ComplEx (Théo, Welbl, Riedel, Gaussier, & Bouchard, 2016) from the tensor factorization approach, ConvKB (Quoc,

Nguyen, Nguyen, & Phung, 2018) and ConPoly model (khobragade, Ghumbre, & Wangde, 2023) from the convolutional approach.

4.3 *Evaluation Metrics*

MNN calculates the score of all the relation types $r \in |R|$ for the provided triplet entity pair $(h, t \in E)$ from the test set $\mathcal{T}$. These scores are then ranked from highest to lowest. Mean Rank (MR↓) averages the rank position of the true relation type, Mean Reciprocal Rank (MRR↑) is reciprocal of the rank position of the true relation type, and Hits@n checks how often the correct relation type appears within the top n ranked positions for each test triplet.

4.4 *Implementation*

We train the baseline and our model with 250 embedding feature vector sizes, 1000 epochs with 512 batch sizes, and optimize by Adam with a learning rate of 0.0001 for fair evaluation on both datasets. Besides, we applied Xavier's for initialization and L_1 for normalization of the vector, the margin set to 1 in the loss function. The dynamic negative sampling strategy (Khobragade, Mahajan, Langi, Mundhe, & Ghumbre, 2022) used for negative triplet for each positive triplet. MNN input layer concatenate head and tail entity vectors of size 500, consisting of 5 hidden layers of size [419, 338, 257, 176, 95], and output layer size is equal to 11 for WN18RR, whereas for FB15k-237 hidden layer size is [457, 414, 371, 328, 285] and output layer size is 237. The sizes of five hidden layers were set according to Equations 5 and 6. The ReLU activates each hidden layer with the dropout 0.3, and the output layer is activated by sigmoid.

5. Results and Discussions

The values in bold case denote the overtaking results. LNAEE overperforms against all the baselines on all evaluation indicators on both datasets. Table 2 illustrates the performance of LNAEE on WN18RR dataset. Specifically, LNAEE achieves 15% comparative increase in Hit@1 over the second-best baseline ConvKB. Similarly, 13% and 1% relative improvements in Hit@3 and Hit@10 respectively. Table 3 illustrates results on FB15k-237. Particularly, LNAEE achieves 10% relative improvement in Hit@1 the second-best baseline. In Hit@3 achieves 8% and in Hit@10 achieves 0.3% relative improvements. The MRR achieves 13% on WN18RR and 7% on FB15k-237 relative improvements over the second-best baseline

ConvKB. All the models perform better on the FB15k-237 because it has a higher entity density than WN18RR. We record the MRR and Hit@1 during the training process for all the models. LNAEE attains MRR between the range of 0.5 to 0.9 and achieves the best result under the initial 300 epochs, shown in Figure 3(a), then saturate. The attention mechanism gave additional assistance to learn the entity embedding effectively. Figure 3(b) shows Hit@1 versus epochs, where LNAEE gains the best result under 300 epochs.

Figure 4 illustrates the effect of increasing the embedding size from 50 to 250 on the LNAEE performance. The results reveal that embedding size is directly proportional to performance in MR decreases, whereas Hit@1 increases gradually. However, the performance up to 150 dimensions increased progressively, but afterward, the performance was saturated with minor improvements. With the help of this result, we decided to evaluate our model with baselines on an embedding feature vector size of 250.

Table 2

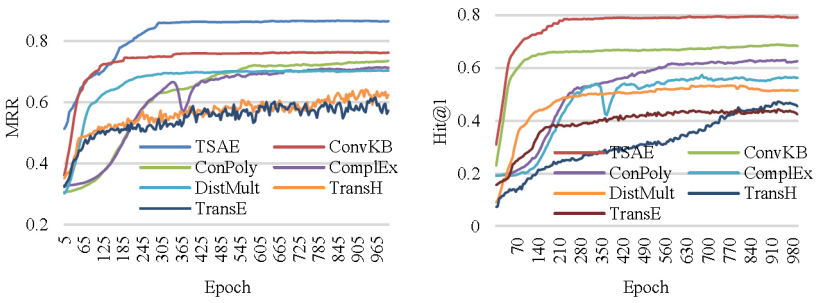
Relation prediction result on the WN18RR dataset

Models	MRR	Hit@1	Hit@3	Hit@10
TransE	0.614	0.439	0.623	0.878
TransH	0.637	0.470	0.659	0.895
DistMult	0.702	0.539	0.727	0.960
ComplEx	0.713	0.571	0.769	0.972
ConPoly	0.730	0.629	0.797	0.982
ConvKB	0.761	0.688	0.841	0.981
LNAEE	**0.864**	**0.793**	**0.955**	**0.992**

Table 3

Relation prediction result on the FB15k-237 dataset

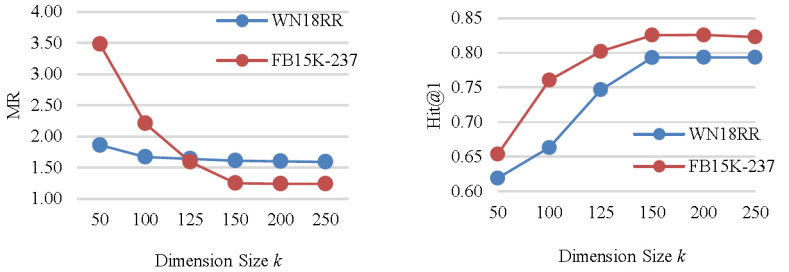
Models	MRR	Hit@1	Hit@3	Hit@10
TransE	0.631	0.534	0.608	0.891
TransH	0.656	0.576	0.653	0.917
DistMult	0.737	0.639	0.705	0.920
ComplEx	0.749	0.659	0.740	0.921
ConPoly	0.832	0.689	0.852	0.945
ConvKB	0.864	0.748	0.897	0.989
LNAEE	**0.931**	**0.825**	**0.977**	**0.998**



(a) (b)

Figure 3

Results of (a) MRR and (b) Hit@1 on the validation set of WN18RR recovered during the training epoch



(a) (b)

Figure 4

Illustration of LNAEE sensitivity in (a) MR and (b) Hit@1 to the embedding dimension size k on WN18RR and FB15k-237 dataset

6. Conclusions

This paper introduces LNAEE, a novel embedding model for completing KG. LNAEE captures local triplet proximity via the skip-gram method and considers the influence of nearby entities using GAT to improve its predictions. A MNN then estimates the most likely relation types between the given entity pair. Experiments on standard datasets demonstrate that LNAEE outperforms existing methods across all rank-based metrics. We have also explored the sensitivity of the model to dimension size. While our research shows promise for relation prediction, there's room for further exploration. Specifically, future research should focus on assessing our model's performance in entity classification and

refining the attention mechanism to effectively capture multi-hop subgraph structures.

References

[1] A. Khobragade and S. Ghumbre, Study and analysis of various link predictions in knowledge graph: A challenging overview, *Intelligent Decision Technologies*, 16. 4, 653-663, (2022).

[2] H. Nguyen, et al., Design an ontology model for intelligent querying system in mathematics education, *Journal of Inter. Math.*, 26.3, 449-473, (2023).

[3] R. C. Poonia, et al., "Face Recognition System for Medical Information Modeling Using Machine Learning." In Handbook of Research on Mathematical Modeling for Smart Healthcare Systems, pp. 353-373. IGI Global, (2022).

[4] A. Kumar, et al., "An enhanced quantum key distribution protocol for security authentication." J*ournal of Discrete Mathematical Sciences and Cryptography* 22, no. 4 (2019).

[5] A. Kumar, et al., "An improved quantum key distribution protocol for verification." *Journal of Discrete Mathematical Sciences and Cryptography* 22, no. 4 (2019).

[6] A. Bordes, et al., Translating embeddings for modeling multi-relational data, In Proceeding of Advances in NLPS, (2013).

[7] W. Zhen, J. Zhang, J. Feng and Z. Chen, Knowledge graph embedding by translating on hyperplanes, In Proceedings of the AAAI, (2014).

[8] L. Yankai, et al., Learning entity and relation embeddings for knowledge graph completion, In Proceedings of the AAAI, (2015).

[9] J. Guoliang, S. He, L. Xu, K. Liu and J. Zhao, Knowledge graph embedding via dynamic mapping matrix, In Procedings of the ACL, (2015).

[10] D. Song, et al., DTransE: Distributed translating embedding for knowledge graph, IEEE Trans. on Para. and Distr. Syst., 32.10, 2509-2523, (2021).

[11] Y. Bishan, et al., Embedding entities and relations for learning and inference in knowledge bases, In Procceding of ICLR, (2015).

[12] T. Théo, et al., Complex embeddings for simple link prediction, In Proceeding of the ICML, (2016).

[13] N. Maximilian, L. Rosasco and T. Poggio, Holographic embeddings of knowledge graphs, In Proceedings of the AAAI, (2016).

[14] D. Tim, et al., Convolutional 2d knowledge graph embeddings, In Proceedings of the AAAI, (2018).

[15] N. D. Quoc, et al., A novel embedding model for knowledge base completion based on convolutional neural network, In Proceeding of NAACL, (2018).

[16] J. Xiaotian, Q. Wang and B. Wang, Adaptive convolution for multi-relational learning, In Proceedings of NAACL, (2019).

[17] A. khobragade, S. Ghumbre and F. Wangde, Convolutional neural network for knowledge graph completion, AIP Conference Proceedings, (2023).

[18] M. Tomas, et al., Distributed representations of words and phrases and their compositionality, In Proceeding of the Adance NLPS, (2013).

[19] A. Khobragade, et al., Effective negative triplet sampling for knowledge graph embedding, *Jour. Info. and Optim. Sci.*, 43.8, 2075-2087, (2022).