SUBGRAPH REPRESENTATION LEARNING WITH HARD NEGATIVE SAMPLES FOR INDUCTIVE LINK PREDICTION

Heeyoung Kwak [⋆]   *Hyunkyung Bae* [⋆]   *Kyomin Jung* [⋆†]

[⋆] Dept. of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[†] Automation and Systems Research Institute, Seoul National University, Seoul, Korea

ABSTRACT

The inductive link prediction in knowledge graphs (KGs) is often addressed to induce logical rules that capture entity-independent relational semantics. Recent studies suggest graph representation learning to encode these logical rules within the local subgraph structures. With this approach, the model can have the inductive ability to cope with unseen entities, which is practical for evolving nature of real-world KGs. However, despite the importance of a high-quality negative sample in link prediction, there is currently no method for selecting hard negatives for inductive link prediction. To overcome this limitation, we propose a new sampling method for selecting hard negative samples given a positive triplet. We also propose Subgraph Infomax (SGI), a novel inductive link prediction model, with a newly-proposed training objective that maximizes the mutual information (MI) between the target relation and the enclosing subgraph. We select hard negative samples by using the pre-trained MI estimator of SGI. The model is then fine-tuned using the selected hard negative samples. Empirically, we demonstrate superior performances of our model on multiple datasets of the inductive KGC benchmark, showing the enhanced connectivity between the target relation embedding and the subgraph representation.

***Index Terms*—** knowledge graph completion, inductive link prediction, graph neural network, deep infomax

1. INTRODUCTION

Knowledge Graphs (KGs) have been very useful for many information retrieval (IR)-related tasks such as query answering, entity linking, and knowledge-augmented question answering. However, due to the incompleteness problem of the KGs, there has been an increasing interest in Knowledge Graph Completion (KGC) task. KGC is a link prediction task to predict missing edges in KG, and can be seen as a question of whether a given triplet is valid or not. Especially, the evolving nature of KGs has led to active research on inductive link prediction, where one needs to make inferences on triplets with entities that are not seen during training.

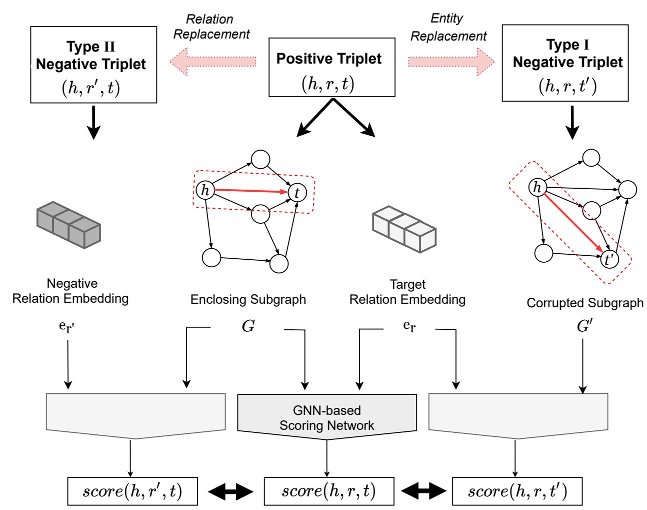

Fig. 1. Two types of negative samples for inductive link prediction models. The score of a given triplet is computed using the GNN-based scoring network where the subgraph representation and the relation embedding are fed.

Recently, many inductive KGC studies have tried inducing the logical rules contained in a local subgraph to learn entity-independent semantics [1, 2, 3, 4]. Since Zhang & Chen [5] have theoretically proven that the enclosing subgraphs surrounding the target triplets are informative for link prediction, several subsequent studies have suggested graph representation learning to encode the logical rules within the local subgraphs [2, 3, 4]. These works estimate the likelihood of a given triplet using the scoring network where the learned subgraph representation and the relation embedding are fed.

Like many conventional link prediction models, these models are trained by contrasting the scores of positive and negative triplets. Generally, they aim to minimize a margin-based ranking loss so that the scores of the positive triplets are larger than those of the negative triplets by a fixed margin. Here, the existing triplets in the given KGs are considered positive triplets. On the other hand, the negative triplets are

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2021R1A2C2008855).

generated by corrupting the positive triplets, such as replacing either the head or tail of positive triplets with another entity or changing only the relation type. We will call the former type of negative sampling as *Type I* Negative Sample (NS) and the latter as *Type II* NS. The two methods of creating NS and the training scheme for link prediction according to those methods are described in Figure 1.

Previously, uniform sampling, a method of randomly selecting either head or tail entities, was frequently used to select *Type I* NS. However, recent literature has questioned whether this naive uniform sampling is informative enough [6, 7]. For example, if we use the uniform sampling to corrupt the tail entity of a positive triplet *(wheat, hypernym, grain)*, it is likely to result in a ridiculous triplet like *(wheat, hypernym, semiconductor)*. Since such easy negative samples produce zero loss as training progresses, several works present a sampling method to generate hard examples. Structure-aware negative sampling (SANS), suggested in [8], constructs hard negative examples simply by selecting a node's k-hop neighborhood, which is less expensive to train compared to GAN-based methods such as [9, 6]. However, in the case of *Type II* NS, there are no such methods to obtain hard negative samples.

As relation sampling has been used to create negative candidates as well as entity sampling, (1) we suggest a hard negative sampling method to create a *Type II* NS that is difficult to distinguish from an original target relation. (2) We also suggest our novel inductive link prediction model, called **Subgraph Infomax (SGI)**, where the relation embedding is trained to contain more meaningful information about subgraphs via the mutual information (MI) maximization objective. Specifically, SGI consists of a GNN-based scoring network for computing the score of a given triplet and a module for MI maximization. We pre-train SGI to maximize the MI between the relation embedding and the subgraph representation. Then we select a hard *Type II* NS by selecting hard relations for the discriminator to distinguish from the target relation. Finally, we fine-tune the scoring network with the selected hard negative samples. We perform ranking evaluation of previously presented SOTA models, GraIL [2], TACT [3], and our model SGI in link prediction task. Compared to the previous models, our model SGI shows superior performance in both inductive versions of Nell-995 and FB15k-237 KG datasets.

2. METHODS

2.1. Subgraph Infomax: SGI

In this section, we provide an overview of our SGI model. The KGC task is to score a triplet (h, r, t) to estimate the probability of a relation r between a head entity h and a tail entity t. Similar to GraIL, we extract an enclosing subgraph $\mathcal{G}_{(h,r,t)}$ around the target nodes, h and t, and use the subgraph structure to score a triplet independently of the node embeddings. In particular, the enclosing subgraph is extracted by intersecting two subsets of k-hop neighboring nodes of the head or tail. The shortest distance of the nodes to the head or tail is used as node features, known as double-radius vertex labeling.

SGI first summarizes the subgraph through a GNN encoder and computes the triplet's score using the encoded-subgraph representation and relation embedding. At the same time, the MI estimator estimates the MI between the relation embedding and the subgraph representation. During the pre-training, the whole scoring network is trained along with the MI estimator. At this stage, we use SANS, which are the hard *Type I* NS, to train the scoring network. Then, we use the trained MI estimator to select a hard *Type II* NS. At the stage of fine-tuning, we fine-tune the scoring network using the selected hard *Type II* NS. The overall process is depicted in Figure 2.

2.2. GNN-based Scoring Network

We use multi-relational R-GCN [10], a GNN-based method designed for modeling multi-relational data, to obtain a subgraph-level representation. The embedding of a node i in the kth layer is given by:

$$\mathbf{h}_i^k = \mathrm{ReLU}(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k \mathbf{W}_r^k \mathbf{h}_j^{k-1} + \mathbf{W}_{self}^k \mathbf{h}_i^{k-1}), \quad (1)$$

where the first term is the aggregated message from the neighbors $\mathcal{N}_i$ of node i. The initial node representation $\mathbf{h}_i^0$ is a node feature labeled by aforementioned double-radius vertex labeling. $\alpha_{i,j}$ denotes an edge attention weight of the edge (i, j) and is given as a function of the source node i, neighbor node j, relation type r_{ij} of the edge (i, j), and the target relation r.

$$\begin{aligned} \mathbf{s} &= \mathrm{ReLU}(\mathbf{A_1}^k[\mathbf{h}_i^{k-1} \oplus \mathbf{h}_j^{k-1} \oplus \mathbf{e}_r \oplus \mathbf{e}_{r_{ij}}] + \mathbf{b_1}^k), \\ \alpha_{ij} &= \sigma(\mathbf{A_2}^k \mathbf{s} + \mathbf{b_2}^k), \end{aligned} \quad (2)$$

where $\mathbf{e}_r$ and $\mathbf{e}_{r_{ij}}$ are relation embeddings for relation r and r_{ij}. Different from GraIL that uses attention relation embeddings, separate from the relation embeddings, we use the same relation embeddings during computing attention weights. This allows the GNN encoder to learn a subgraph representation with enhanced connectivity between the subgraph representation and the input relation embedding through the MI estimator that will be described later.

After L layers of message passing, we obtain a subgraph-level representation $\mathbf{h}_{\mathcal{G}_{(h,r,t)}}$ by concatenating the node representation of the source, target, and the average-pooled representation of all the node:

$$\mathbf{h}_{\mathcal{G}_{(h,r,t)}} = \mathbf{h}_h^L \oplus \mathbf{h}_t^L \oplus \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbf{h}_i^L, \quad (3)$$

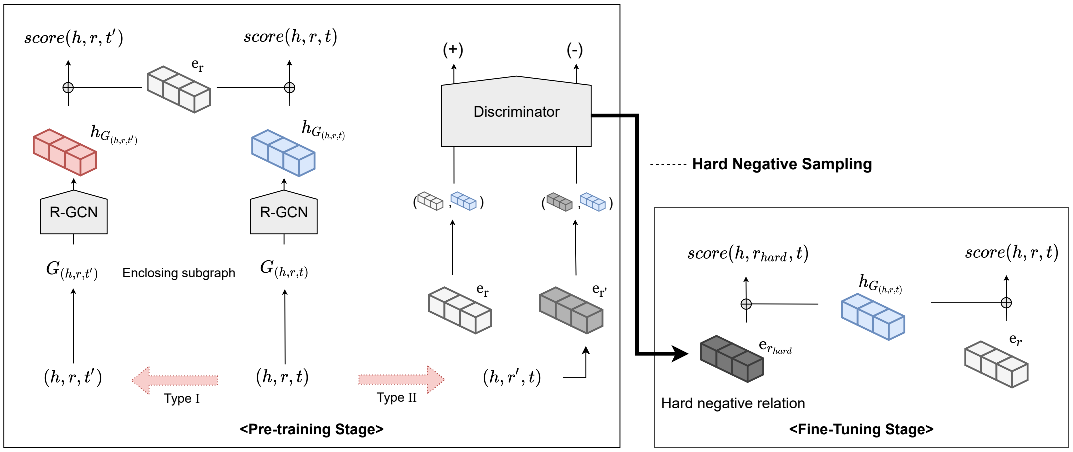

Fig. 2. An overview of SGI learning scheme. During pre-training, SGI is trained with hard *Type I* examples, and the discriminator is trained to maximize the MI. Then, hard *Type II* examples selected with the discriminator are used for fine-tuning.

where $\mathcal{V}$ denotes the set of vertices in $\mathcal{G}_{(h,r,t)}$. To make the model's performances robust to the number of GNN layers, we adopt JK-connections [11] as in GraIL. Finally, we compute a score for (h, r, t) using the subgraph representation and the target relation embedding as follows:

$$\text{score}(h, r, t) = \mathbf{W}[\mathbf{h}_{\mathcal{G}_{(h,r,t)}} \oplus \mathbf{e}_r] \tag{4}$$

2.3. Subgraph-Relation MI Maximization

The key idea behind our approach is to strengthen the connectivity between the enclosing subgraph around the target triplet and the target relation. Based on the assumption that the enclosing subgraphs contain the logical rules related to the target triplet, we implement the objective to maximize the MI between the subgraph representation encoded with an R-GCN model and the relation embedding.

Motivated by previous works [12, 13] on using MI estimator for graph representation learning, we employ a discriminator $\mathcal{D}_\psi(\mathbf{h}_{\mathcal{G}_{(h,r,t)}}, \mathbf{e}_r)$ that represents the feasibility of the subgraph-relation pair. ψ refers to the parameters of $\mathcal{D}$, which is a bilinear function in our study. Negative samples for subgraph-relation pair are given as $(\mathbf{h}_{\mathcal{G}_{(h,r,t)}}, \mathbf{e}_{r'})$, where $\mathbf{h}_{\mathcal{G}_{(h,r,t)}}$ is the graph representation from the positive triplet and $\mathbf{e}_{r'}$ is the relation embedding for a random negative relation. This can also be seen as a simple form of *Type II* NS. We use a noise-contrastive type objective with a binary-cross entropy loss suggested in [14] for MI maximization so the estimated MI on subgraph-relation pairs over the training set $\mathcal{G}_{train}$ is given as follows:

$$\begin{aligned}\mathcal{I}_{\phi,\psi} := \sum_{(h,r,t)\in\mathcal{G}_{train}} \log[\mathcal{D}_\psi(\mathbf{e}_r, \mathbf{h}_{\phi,\mathcal{G}_{(h,r,t)}})] + \\ \sum_{(h,r,t)\in\mathcal{G}_{train}, r'\in\mathcal{R}, r'\neq r} \log[(1 - \mathcal{D}_\psi(\mathbf{e}_{r'}, \mathbf{h}_{\phi,\mathcal{G}_{(h,r,t)}}))],\end{aligned} \tag{5}$$

where $\mathcal{I}_{\phi,\psi}$ is the MI estimator modeled by discriminator $\mathcal{D}_\psi$ and ϕ denotes the set of parameters of a R-GCN encoder. By maximizing the above MI objective, we train the whole networks to learn the subgraph representation that is strongly connected to the target relation embedding.

2.4. Training Objective

During the pre-training stage, we use binary-cross entropy (BCE) loss to discriminate the positive and negative triplets as follows:

$$\mathcal{L} = \sum_{p_i\in\mathcal{G}_{train}} -\log(\text{score}(p_i)) - \log(1 - \text{score}(n_i)), \tag{6}$$

where p_i is the positive triplet and n_i is the negative triplet extracted by using SANS strategy. We will mark this NS as *Type I* (`SANS`). Combined with the MI objective in (5), the total loss function for pre-training is defined by:

$$\mathcal{L}_{p.t.} = \mathcal{L} - \mathcal{I}_{\phi,\psi}. \tag{7}$$

After pre-training, we select *Type II* (`hard`) using the pre-trained MI estimator. Given a target relation r, we simply select the relation r' with the largest discriminative loss $\mathcal{D}_\psi(\mathbf{e}_{r'}, \mathbf{h}_{\phi,\mathcal{G}_{(h,r,t)}})$. The final loss for fine-tuning is given as:

$$\begin{aligned}\mathcal{L}_{f.t.} &= \sum_{p_i\in\mathcal{G}_{train}} [-\log(\text{score}(p_i)) - \log(1 - \text{score}(n_i^{hard}))], \\ n_i^{hard} &= (h, \operatorname*{argmax}_{r'\neq r} \mathcal{D}_\psi(\mathbf{e}_{r'}, \mathbf{h}_{\phi,\mathcal{G}_{(h,r,t)}}), t).\end{aligned} \tag{8}$$

Table 1. Inductive link prediction results evaluated on randomly selected negative triplets

Dataset		NELL-995 v1			FB15k-237 v1		
Model	Types of Negative Sample	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
GraIL	I (`random`)	52.04	46.50	59.50	48.56	41.46	64.15
GraIL	I (`SANS`)	12.17	17.54	22.13	45.49	37.80	56.83
GraIL	II (`random`)	43.99	33.12	50.89	33.39	22.26	58.73
GraIL	I (`SANS`) + II (`random`)	41.17	35.71	54.56	34.20	24.04	55.01
TACT	II (`all`)	16.43	11.40	21.50	20.26	14.64	26.10
SGI (Pre-train)	I (`SANS`) + II (`random`)	75.09	**70.38**	82.69	49.80	42.17	63.00
SGI (Fine-tune)	II (`hard`)	**75.76**	69.79	**84.51**	**51.50**	**44.51**	**65.44**

Table 2. Inductive link prediction results evaluated on SANS negative triplets

Dataset		NELL-995 v1			FB15k-237 v1		
Model	Types of Negative Sample	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
GraIL	I (`random`)	45.09	39.07	52.62	35.84	27.56	49.76
GraIL	I (`SANS`)	21.08	7.79	36.40	44.06	36.10	56.34
GraIL	II (`random`)	52.54	45.35	76.81	41.03	35.87	51.22
GraIL	I (`SANS`) + II (`random`)	40.63	35.54	49.98	40.99	32.14	52.35
TACT	II (`all`)	20.20	14.60	26.40	18.32	12.29	25.76
SGI (Pre-train)	I (`SANS`) + II (`random`)	71.28	64.59	**83.82**	43.30	32.73	66.16
SGI (Fine-tune)	II (`hard`)	**74.07**	**68.69**	83.14	**48.86**	**39.94**	**66.36**

3. EXPERIMENTS

3.1. Datasets

We use the same benchmark datasets for inductive link prediction as in GraIL [2]. The train set and the test set have no overlapping entities in these datasets. Here, we report the experimental results of Nell-995 v1 and FB15k-237 v1. These datasets are the version with the smallest number of links in training set among the four version introduced in [2].

3.2. Training and Evaluation

We follow the hyperparameters optimized in GraIL, where a 3-layer GNN is employed with the dimension of all latent representation i.e., node and relation embeddings equal to 32. We use Adam optimizer with an initial learning rate of 0.01 for pre-training, 1e-4 for fine-tuning, L2 penalty of 5e-4, and batch size of 16. We train the models on a GTX 1080 Ti for 50 epochs with early stopping.

In link prediction, one aims to predict the other entity given another entity and the relation type, i.e., predicting the tail given $(h, r, ?)$ or predicting the head given $(?, r, t)$. We evaluate the models on Mean Reciprocal Rank (MRR), Hits at 1 (H@1), and H@10, by ranking each validation triplet among 50 other negative candidates. As for the negative triplets, the head entity (or tail) is sampled using both uniform sampling and SANS.

3.3. Results

The experimental results on all models are reported in table 1 and 2. All results are averaged after measuring five times. GraIL was originally trained with NS extracted by uniform sampling, which we call *Type I* (`random`) but we train GraIL with all other types of NS for fair comparison. However, We train TACT as in the original work, using *Type II* NS, which is made using all relations except target relations. We mark this type as *Type II* (`all`). As a result, SGI showed superior performances when evaluated on both randomly selected and SANS negatives, showing the effect of MI maximization. It showed higher performances when fine-tuned, confirming that the model becomes robust when trained with high-quality hard negatives.

4. CONCLUSIONS

We present a novel approach for inductive link prediction for KGs. We propose a new model, called SGI, that maximizes the MI between the target relation and its enclosing subgraph. We also propose a new sampling method for selecting hard NS by using the pre-trained MI estimator of SGI. Our model shows superior performances on two inductive KGC benchmarks, showing the enhanced connectivity between the target relation embedding and the subgraph representation. Higher performances of the fine-tuned SGI demonstrates the qulity of hard negatives extracted by our method.

5. REFERENCES

[1] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang, "DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs," in *Advances in Neural Information Processing Systems*, 2019, vol. 32.

[2] Komal K. Teru, Etienne Denis, and William L. Hamilton, "Inductive relation prediction by subgraph reasoning.," *arXiv: Learning*, 2020.

[3] Jiajun Chen, Huarui He, Feng Wu, and Jie Wang, "Topology-Aware Correlations Between Relations for Inductive Link Prediction in Knowledge Graphs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021.

[4] Sijie Mai, Shuangjia Zheng, Yuedong Yang, and Haifeng Hu, "Communicative Message Passing for Inductive Relation Reasoning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021.

[5] Muhan Zhang and Yixin Chen, "Link prediction based on graph neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 5165–5175.

[6] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *International Conference on Learning Representations*, 2019.

[7] Liwei Cai and William Yang Wang, "Kbgan: Adversarial learning for knowledge graph embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1470–1480.

[8] Kian Ahrabian, Aarash Feizi, Yasmin Salehi, William L Hamilton, and Avishek Joey Bose, "Structure aware negative sampling in knowledge graphs," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6093–6101.

[9] Peifeng Wang, Shuangyin Li, and Rong Pan, "Incorporating gan for negative sampling in knowledge representation learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[10] Michael Sejr Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling, "Modeling relational data with graph convolutional networks," in *ESWC*, 2018.

[11] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds. 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462, PMLR.

[12] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm, "Deep Graph Infomax," in *International Conference on Learning Representations*, 2019.

[13] Fan-Yun Sun, Jordan Hoffman, Vikas Verma, and Jian Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *International Conference on Learning Representations*, 2019.

[14] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio, "Learning deep representations by mutual information estimation and maximization," in *International Conference on Learning Representations*, 2019.