

Relation-aware Graph Attention Model With Adaptive Self-adversarial Training

Xiao Qin^{1*}, Nasrullah Sheikh^{1*}, Berthold Reinwald¹, Lingfei Wu²

¹IBM Almaden Research Center, ²IBM Thomas J. Watson Research Center
{xiao.qin, nasrullah.sheikh}@ibm.com, {reinwald, wuli}@us.ibm.com

Abstract

This paper describes an end-to-end solution for the relationship prediction task in heterogeneous, multi-relational graphs. We particularly address two building blocks in the pipeline, namely heterogeneous graph representation learning and negative sampling. Existing message passing-based graph neural networks use edges either for graph traversal and/or selection of message encoding functions. Ignoring the edge semantics could have severe repercussions on the quality of embeddings, especially when dealing with two nodes having multiple relations. Furthermore, the expressivity of the learned representation depends on the quality of negative samples used during training. Although existing hard negative sampling techniques can identify challenging negative relationships for optimization, new techniques are required to control false negatives during training as false negatives could corrupt the learning process. To address these issues, first, we propose RelGNN— a message passing-based heterogeneous graph attention model. In particular, RelGNN generates the states of different relations and leverages them along with the node states to weigh the messages. RelGNN also adopts a self-attention mechanism to balance the importance of attribute features and topological features for generating the final entity embeddings. Second, we introduce a parameter free negative sampling technique – adaptive self-adversarial (ASA) negative sampling. ASA reduces the false negative rate by leveraging positive relationships to effectively guide the identification of true negative samples. Our experimental evaluation demonstrates that RelGNN optimized by ASA for relationship prediction improves state-of-the-art performance across established benchmarks as well as on a real industrial dataset.

1 Introduction

Modern data are often relational in nature with heterogeneous interactions between entities and complicated dependency structures. Knowledge graphs (Ji et al. 2020) (KGs), as a prime example of heterogeneous graphs, model real-world objects, events or concepts as well as various relations among them. Making sense of data in such form has gained tremendous attention in large enterprises as graph insights enable new capabilities or services. In particular, this

paper focuses on the task of relationship prediction, i.e. predicting the types of relations between entities with rich attributes where the entities are nodes in a graph and the existing relationships form the edges between them. We propose two building blocks in an end-to-end solution for this task, namely *heterogeneous graph representation learning with relation-aware attention* and *adaptive self-adversarial negative sampling* for model training.

First, while knowledge graph embedding methods (Ji et al. 2020; Yang et al. 2015; Sun et al. 2019; Hayashi and Shimbo 2017; Dettmers et al. 2018) have achieved great success for relationship prediction for heterogeneous graphs, there has been an increasing interest in leveraging Graph Neural Networks (GNNs) for this task (Schlichtkrull et al. 2018). While early research on GNNs mainly focuses on homogeneous graphs (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Wu et al. 2019; Chen, Wu, and Zaki 2020a,b), more recent variations and extensions (Schlichtkrull et al. 2018; Wang et al. 2019; Cen et al. 2019; Dong, Chawla, and Swami 2017; Zhang et al. 2019a; Hu et al. 2020; Chen, Wu, and Zaki 2020c) cope with the heterogeneity aspect of the nodes and edges. The main idea of using GNNs for heterogeneous graphs is to learn dedicated, type-aware encoding functions for information propagation and aggregation (Yang et al. 2020a). However, existing research tends to ignore the semantics of the edges, that is the edge information is only used either for graph traversal (e.g., meta-path based neighborhood sampling (Dong, Chawla, and Swami 2017)) and/or selection of encoding functions (Schlichtkrull et al. 2018). Ignoring the semantics of edges during aggregation decreases the expressivity of the node embeddings as messages propagated even from the same source may contribute quite differently to its neighbor’s embedding. For example, a buyer entity can have multiple relations with a product such as “purchase”, “has maintenance contract”, and “has warranty”. Besides the difference of messages sent by this product via multiple relations, these messages may not be equally important to characterize the buyer. Existing GNNs, message aggregation methods in particular, tend to ignore such phenomenon missing out on the opportunity to obtain more expressive node embeddings. The key challenge is to design a mechanism for the aggregation function to generate differentiable signals from both heterogeneous neighbors and edges.

* authors contributed equally.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Second, most graph representation learning methods can be unified within a *sampled noise contrastive estimation* framework (Yang et al. 2020b), especially for link/relationship prediction task. Random negative sampling has been widely adopted for its simplicity and efficiency. However, it suffers seriously from the *vanishing gradient problem* as most negative samples can be easily classified (Zhang et al. 2019b). Generative adversarial network (GAN) based solutions (Cai and Wang 2018; Wang, Li, and Pan 2018) are later proposed for addressing this issue. Using GAN, the generator acts like a negative sampler to identify challenging relationships for the discriminator to learn. However, GANs have more parameters and are generally difficult to train.

Recently, a line of research exploits the model itself (Sun et al. 2019; Zhang et al. 2019b) to identify the challenging negative samples. These models work under the assumption that the hardness of a negative sample is proportional to the error it causes. Despite their superior performance, they still struggle with the problem of over-training on *false negatives* as the training progresses. Thus, the negative samples that cause extremely high errors would not be trustworthy especially in missing link/relationship prediction scenarios. How to balance the exploitation of hard negative samples, and avoiding false negatives becomes a challenging problem since a clear cut boundary between them does not exist. Moreover, the decision boundaries could vary from case to case due to the heterogeneity of the graphs failing the simple uniform thresholding approaches (Ying et al. 2018; Sun et al. 2019; Zhang et al. 2019b).

Our Approach. To address the above issues, we propose RelGNN— a graph attention model for learning entity embeddings in heterogeneous graphs and a negative sampling algorithm called adaptive self-adversarial (ASA) negative sampler for model training.

RelGNN follows the *spatial approach* (Wu et al. 2019) based on the *message passing paradigm* (Battaglia et al. 2018) in generating embeddings of entities in an *inductive* manner (Hamilton, Ying, and Leskovec 2017). RelGNN starts with the attribute embedding layers configured for different node type. It consists of multiple graph convolution layers and the messages collected from the neighbors are encoded differently according to their edge type. Besides, the messages are weighed via a multi-head attention mechanism which considers the states of the nodes on both ends of the edge as well as the state of the edge since two entities could have multiple relations in the heterogeneous graph. To balance the importance of the attribute features and the topological features for the relationship prediction, the final embedding combines the attribute embedding and the output of the message propagation using a self-attention mechanism.

To capture the dynamic distribution of the negative samples during training, ASA algorithm makes use of the trained model from the prior iteration to evaluate the hardness of a generated negative sample, i.e. the higher the gradient it causes, the harder the sample is to the model. To overcome the false negative issue, ASA also evaluates and considers the hardness of the positive samples. The idea, inspired by a negative sampling method for training dialogue systems (Li et al. 2019), is based on the following heuristic: since the

negative samples are derived from an existing relationship, the level of confidence of the model on the positive sample should match its level of confidence on the negative one. If the gradient caused by a negative sample is way higher, that particular sample could be a false negative, and therefore should be avoided.

It is worthwhile to highlight the following contributions of this paper:

- We propose RelGNN— a message passing-based graph attention model for heterogeneous graph which leverages the edge semantics to boost the expressive power.
- We propose ASA – a parameter free negative sampling technique that identifies hard negative samples while effectively reducing the false negative rate.
- We evaluate RelGNN and ASA negative sampler by an ablation study and a position study which compares our method against state-of-the-art approaches in relationship prediction task on inductive benchmarks as well as a real industrial dataset.

2 Preliminary

In this section, we define the heterogeneous graph and introduce the main notations. Then, we introduce the message passing framework (Battaglia et al. 2018) upon which we build RelGNN.

2.1 Attributed Heterogeneous Graph

Let $G = (V, E, A, R, \phi)$ denote a heterogeneous graph. $V = \{v_1, \dots, v_n\}$ is a set of nodes in G . $A = \{a_1, \dots, a_k\}$ defines the node attribute schema of G . Each node $v_i \in V$ is associated with a node type $\phi(v_i)$. Its corresponding attribute schema is a subschema of A denoted by $A_{\phi(v_i)} \subseteq A$. $E = \{e_1, \dots, e_m\}$ is a set of directed edges and R defines the edge types in G . Each edge $e_k = (v_i, r_i, v_j)$ indicates a relationship from v_i to v_j of a relation type $r_i \in R$. Note that in this definition, two nodes can have multiple relations. In the rest of the paper, the terms node and entity, edge and relationship are used interchangeably.

2.2 Message Passing-based GNN

A message passing-based GNN (Battaglia et al. 2018) adopts a neighborhood aggregation strategy where the representation of a node is learned in an iterative manner by aggregating representations of its neighbors. After t iterations (layers) of aggregation, a node’s representation encodes the structural information within its k -hop neighborhood. Formally, the t -th layer of a GNN is:

$$h_{aggv_i}^{(t)} = \text{Aggregate}^{(t)}(\{h_{v_j}^{(t-1)} | v_j \in \text{Neighbor}(v_i)\}), \quad (1)$$

$$h_{v_i}^{(t)} = \text{Combine}^{(t)}(h_{v_i}^{(t-1)}, h_{aggv_i}^{(t)}), \quad (2)$$

where $h_{v_i}^{(t)}$ is the vector representation of the node v_i at the t -th iteration. In inductive learning setting, the node state is usually initialized by the node attributes embedding. The exact definitions of $\text{Neighbor}(\cdot)$, $\text{Aggregate}^{(t)}(\cdot)$

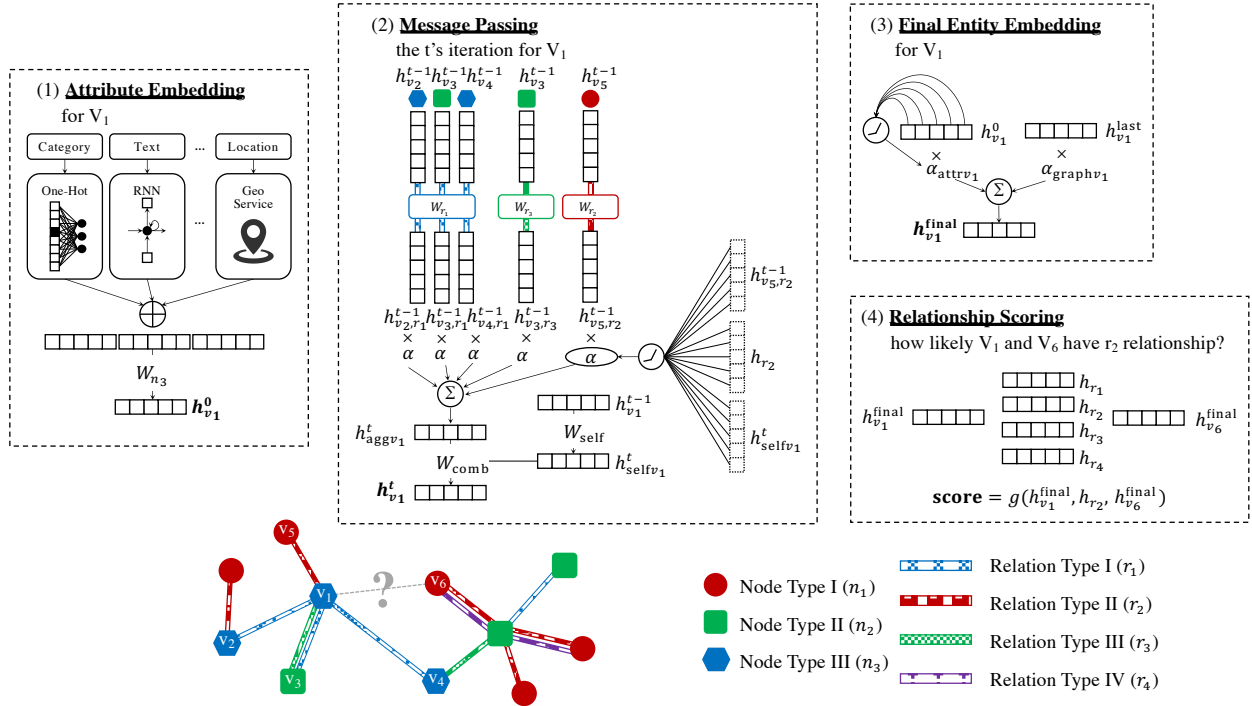


Figure 1: Illustration of RelGNN. (1) an attribute embedding network for a specific node type. (2) a message propagation process for v_1 at time t . (3) the self-attention mechanism for generating the final embedding of v_1 . (4) relationship scoring.

and $Combine^{(t)}(\cdot)$ in GNNs define their modeling approaches (Xu et al. 2019).

3 Methodology

In this section, we introduce RelGNN— a message passing-based graph attention neural network for generating entity embeddings for a heterogeneous graph, and a new negative sampling method, called ASA to train RelGNN for relationship prediction.

As illustrated in Figure 1, RelGNN consists of four components. The node heterogeneity in the **attribute embedding** phase is dealt by configuring dedicated attribute embedding networks for each node type and eventually bringing the attribute embeddings of different node types into the same space. An attribute embedding network for a specific node type consists of individual embedding layers or methods appropriate for different modalities and types defined by the node attribute schema. For example, the attribute embedding network for v_1 in Figure 1(1) consists of a multilayer perceptron (MLP) network to process categorical attribute in one-hot-encoding, a recurrent neural network-based language model (Mikolov et al. 2011) to process text attribute and a geo-location API to obtain the latitude and longitude of an address. The outputs are concatenated and transformed by a weight matrix to the final attribute embedding of v_i denoted by $h_{v_i}^{(0)}$.

RelGNN learns topological features of nodes through **message passing**. The key idea is to encode and weigh the messages differently by fully incorporating edge hetero-

geneity and node state. We will discuss this in detail in Section 2.2. RelGNN then generates the final entity embedding by combining the node attribute embedding and node graph embedding via a self-attention mechanism.

For relationship prediction, RelGNN adopts factorization approaches (Yang et al. 2015; Hayashi and Shimbo 2017) to compute how likely two nodes share a specific relation by considering the embedding of the two nodes as well as the corresponding relation embedding. Finally, we introduce a new negative sampling method to efficiently train RelGNN in Section 3.2. It can be generalized to train a broad spectrum of GNNs for link/relationship prediction task.

3.1 Attention-based Message Passing Network for Attributed Heterogeneous Graphs

Message Propagation. We start by describing the node embedding generation process in a single message passing iteration. For node v_i , we define a propagation function for calculating the forward-pass update in G :

$$h_{v_i}^{(t)} = \sigma \left(\sum_{r \in R} \sum_{v_j \in N_{v_i}^r} \alpha_{(v_i, v_j)} W_r^{(t-1)} h_{v_j}^{(t-1)} + W_{self}^{(t-1)} h_{v_i}^{(t-1)} \right), \quad (3)$$

where σ is an activation function, $N_{v_i}^r$ denotes the set of immediate neighbors of v_i which connect with v_i via $r \in R$, $W_r^{(t-1)}$ is a weight matrix dedicated for encoding messages sent via a particular relation r , $W_{self}^{(t-1)}$ is a weight matrix that encode v_i 's embedding from the last iteration,

and $\alpha_{(v_i, v_j)}$ is an attention weight for the message sent from v_j . The first term in Equation 3 defines our aggregation function in the message passing network (Equation 1). First, the messages sent via different types of edges are encoded differently by introducing a dedicated weight matrix (Schlichtkrull et al. 2018). To avoid overfitting, we apply *basis-decomposition* on these matrices. Second, the messages are weighed by considering the nodes as well as the relation. It addresses the issue of having multiple relations between two nodes, e.g. v_3 and v_1 in Figure 1, which is typical in heterogeneous graphs. In particular, we use a multi-head attention mechanism to compute the attention coefficient. A single head attention weight in Equation 3 is expressed as:

$$\alpha_{(v_i, v_j)} = \frac{\exp(\sigma(\mathbf{a}_e^\top [W_{self}^{(t-1)} h_{v_i}^{(t-1)} \parallel h_r \parallel W_r^{(t-1)} h_{v_j}^{(t-1)}]))}{\sum_{r' \in R} \sum_{v_n \in N_{v_i}^{r'}} \exp(\sigma(\mathbf{a}_e^\top [W_{self}^{(t-1)} h_{v_i}^{(t-1)} \parallel h_{r'} \parallel W_{r'}^{(t-1)} h_{v_n}^{(t-1)}]))} \quad (4)$$

where \cdot^\top represents transposition, \parallel is the concatenation operation, h_r is the state of the relation r and \mathbf{a}_e is weight vector for a single head attention mechanism. In our experiment, we use the LeakyReLU as the activation function. The above computation is illustrated in Figure 1(2). The final multi-head attention-based propagation is then expressed as:

$$h_{v_i}^{(t)} = \sigma\left(\frac{1}{L} \sum_{l=1}^L \left(\sum_{r \in R} \sum_{v_j \in N_{v_i}^r} \alpha_{(v_i, v_j)}^l W_r^{(t-1)} h_{v_j}^{(t-1)} + W_{self}^{(t-1)} h_{v_i}^{(t-1)} \right)\right), \quad (5)$$

where L is the number of heads.

Final Entity Embedding. Based on our observation, the attribute embedding itself sometimes provides good expressive power to make the relationship prediction, and such cases vary depending upon the particular nodes and edges. Therefore, we use a self-attention mechanism to balance the contribution of the attribute embedding $h_{v_i}^{(0)}$ and the final graph embedding $h_{v_i}^{last}$:

$$h_{v_i}^{final} = \alpha_{attr} h_{v_i}^{(0)} + \alpha_{graph} h_{v_i}^{last}, \quad (6)$$

$$\alpha_{attr} = \frac{\exp(\sigma(\mathbf{a}_s^\top h_{v_i}^{(0)}))}{\exp(\sigma(\mathbf{a}_s^\top h_{v_i}^{(0)})) + \exp(\sigma(\mathbf{a}_s^\top h_{v_i}^{last}))}, \quad (7)$$

where \mathbf{a}_s is a shared attention vector. α_{graph} is computed in the same way as Equation 7. Specifically, we use a multi-head self-attention mechanism and employ *averaging* for aggregating states from different heads.

3.2 Adaptive Self-Adversarial Negative Sampling

Negative sampling (Mikolov et al. 2013) is a technique to approximate a *softmax* function with a huge output layer. It is based on the idea of noise contrastive estimation which says that a good model should differentiate positive signals from negative ones. It has been adapted to graph learning problems (Yang et al. 2020b) especially where the learning objective involves predicting relations between nodes:

$$\operatorname{argmin}_{\theta} \sum_{(v_i, r, v_j) \in E} [\ell(+1, d_r(f(v_i), f(v_j))) + \ell(-1, d_r(f(\bar{v}_m), f(\bar{v}_n)))], \quad (8)$$

where θ denotes the model parameters, ℓ is usually defined as cross entropy, $\bar{v}_m, \bar{v}_n \in V$ are parts of a negative relationship sample, i.e. $(\bar{v}_m, r, \bar{v}_n) \notin E$, f refers to a graph embedding network (RelGNN in our case), and d_r is a scoring function. In this paper, we define d_r as DistMult (Yang et al. 2015) factorization which can be replaced by other *relation learning* models (Hayashi and Shimbo 2017) as long as it learns embeddings for relations:

$$d_r(f(v_i), f(v_j)) = \operatorname{softmax}(h_{v_i}^{final\top} M_r h_{v_j}^{final}), \quad (9)$$

where $M_r \in \mathbb{R}^{d \times d}$ is a diagonal matrix associated with $r \in R$. h_r is the diagonal of M_r , which is used in Equation 4 for computing the attention coefficient.

A negative sample in the context of relationship prediction is normally generated by altering one of the nodes of an existing edge while keeping the relation type fixed, e.g. $(v_i, r, \bar{v}_n) \notin E$ or $(\bar{v}_m, r, v_j) \notin E$. Random negative sampling is commonly adopted due to its simplicity and efficiency. However, (Zhang et al. 2019b) has shown that only a small subset of all possible negative samples are useful for training while most of the negative samples are trivial cases from which the model does not gain much discriminative power.

Recently, GAN-based solutions (Cai and Wang 2018; Wang, Li, and Pan 2018) have attracted a lot of attention for this particular problem. The idea is to use the generator as a negative sampler to generate hard negative relationships for the discriminator. However, GANs introduce more parameters and are generally difficult to train. To overcome the computational barrier, self-adversarial negative sampling has been proposed (Sun et al. 2019; Zhang et al. 2019b). The core idea is to use the model itself to evaluate the hardness of negative samples, i.e., when using the *stochastic gradient descent* algorithm for optimization, a negative sample with a high gradient for the current model is a hard negative sample. Let d'_r and f' be the scoring and embedding function trained from the previous iteration, then for a relationship $(v_i, r, v_j) \in E$, the negative sample selection is defined as:

$$\operatorname{argmax}_{\{\bar{v}_m, r, \bar{v}_n\} \notin E} d'_r(f'(\bar{v}_m), f'(\bar{v}_n)), \quad (10)$$

meaning that the negative sample which is predicted as a positive relationship with a high score is preferred to be used in the current training process. To reduce the computational cost of evaluating every possible negative sample to optimize Equation 10, NSCache (Zhang et al. 2019b) only evaluates a small pool of negative samples randomly selected from the complete set and samples the ones only from this small pool. Although such methods offer superior performance than GAN-based solutions, they do not solve the problem of introducing false negatives because Equation 10 always assigns the highest sampling probability to the worst mispredictions. To tackle this, we propose the adaptive self-adversarial (ASA) negative sampling method. Since a negative sample is a variation of the positive relationship, the

idea is to make use of the positive relationship during the evaluation to control the hardness accordingly. Specifically, we alter the selection strategy from Equation 10 to:

$$\operatorname{argmin}_{\{\bar{v}_m, r, \bar{v}_n\} \notin E} |d'_r(f'(v_i), f'(v_j)) - d'_r(f'(\bar{v}_m), f'(\bar{v}_n)) - \mu|, \quad (11)$$

where μ is a positive constant. Instead of forcing the selection towards the hardest negative sample, ASA selects a negative sample with moderate difficulty by considering the score on the respective positive sample. That is, the score of a selected negative sample may not be higher than the score of the positive relationship from which it is derived. The implication is that the negative sample with score higher than its respective positive relationship can be a false negative. We also introduce a margin μ as a hyperparameter to further control the hardness – the higher the μ the easier the case. Inspired by the self-paced learning (Kumar, Packer, and Koller 2010), we also experiment with different decay functions, e.g. exponential and linear decay function for μ to increase the hardness as the training progresses.

4 Experimental Evaluation

This section summarizes our experimental setup and reports, and analyzes the measurements.

Dataset. We use two established heterogeneous graph datasets, namely Amazon and Youtube (Cen et al. 2019) which come with standard validation and test sets. To evaluate our method in a real world scenario, we use a proprietary enterprise heterogeneous graph dataset Company concerning buy/sell transactions. Company includes challenging validation and test sets by introducing high quality negative samples based on our knowledge of the graph. Basic statistics of these datasets are reported in Table 1.

Metric. We adopt two kinds of evaluation criteria. First, following the common practice, we treat the relationship prediction as a binary classification task, i.e. given two entities and a relation type, predict whether or not such combination holds. We report micro AUC (area under the curve) of ROC (receiver operating characteristic curve) and PRC (precision-recall curve) as well as micro F1 with a fixed cutoff threshold. Second, to evaluate the negative sampling techniques for model training, we report filtered MRR (mean reciprocal rank) and Hit@k. These ranking metrics evaluate how well the model can separate a positive relationship from all possible negative relationships which are generated by altering one of the entities in the positive relationship. Each test case consists of one positive relationship and all the possible negative samples with the same relation type and at least one of the node in the positive relationship. MRR indicates the overall ranking performance of all cases. Hit@k computes the proportion of test cases where the positive relationship appears in the top k results.

Baselines. We compare RelGNN against the state-of-the-art relationship prediction models which can be classified into three categories:

- **Relation learning methods**, widely studied in the context of knowledge graph embedding learn a scoring function which evaluates an arbitrary relationship involving

two entities and a specific relation. **DistMult** (Yang et al. 2015) exploits a similarity-based scoring function. Each relation is represented by a diagonal matrix and the scoring function is defined as a bilinear function. **ComplEx** (Hayashi and Shimbo 2017) extends this idea by introducing the entities and relations into a complex space. **ConvE** (Dettmers et al. 2018) utilizes 2-D convolution over embeddings and multiple layers of nonlinear features to model the interactions between entities and relations.

- **Proximity-preserving methods** capture the topological information by preserving different types of proximity among the entities in the graph. In particular, we compare against a popular random walk-based method **meta-path2vec** (Dong, Chawla, and Swami 2017). It utilizes the paths traversed by meta-path guided random walks to model the context of an entity regarding heterogeneous semantics.
- **Message passing methods**, as discussed in Section 2.2, learn the entity embedding by aggregating the neighbors’ information. **R-GCN** (Schlichtkrull et al. 2018) considers the heterogeneity by having a dedicated aggregation function for each edge type. **HAN** (Wang et al. 2019) utilizes meta-path to model higher order proximity. **GATNE** (Cen et al. 2019) learns multiple embeddings for an entity, each of which encodes the entity w.r.t a specific edge type.

We compare ASA against state-of-the-art negative sampling techniques. Self-adversarial negative sampling strategies (Sun et al. 2019; Zhang et al. 2019b) have shown a superior performance than generative adversarial network (GAN)-based solutions (Cai and Wang 2018; Wang, Li, and Pan 2018) both in terms of effectiveness and efficiency. We choose the latest variation of the self-adversarial negative sampling method **NSCaching** as the baseline. It provides some flexibility of avoiding the false negative which will be discussed in Section 4.1. For simplicity, we refer to the combination of RelGNN and ASA as RelGNN and use \pm ASA to denote whether a model is optimized by ASA.

4.1 Position Study

In this set of experiments, we evaluate and compare RelGNN against state-of-the-art methods.

Main Result. We first evaluate the performance of RelGNN on the benchmarks. Following the same fashion, we select some negative samples for the validation and test partitions of the Company dataset as well. The task is to predict whether or not a given relationship is true. The overall performance of different methods are reported in Table 2.

Overall, RelGNN (RelGNN trained with ASA negative sampling) performs better than the message passing baselines as well as the proximity preserving baseline. The relation learning methods which do not consider the topological features have lower measurements among all. RelGNN achieves the best AUC and AP on the benchmarks and has the best F1 as well on the Company dataset. We use a fixed threshold 0.5 for computing F1 as it seems natural if one interprets the score as a probability. Although the potential best F1 a model can achieve is reflected by AP, showing the F1 with a uniform threshold demonstrates

Dataset	Nodes	Edges	Attr. Dim	Relations	(train/valid/test)
Amazon (Cen et al. 2019)	10,166	148,865	1,156	2	85/5/10
Youtube (Cen et al. 2019)	2,000	1,310,617	2,000	5	85/5/10
Company	11,585	90,262	775	6	80/10/10

Table 1: Statistics of the datasets.

Methods	Amazon			Youtube			Company		
	AUC \uparrow	AP \uparrow	F1 \uparrow (0.5)	AUC \uparrow	AP \uparrow	F1 \uparrow (0.5)	AUC \uparrow	AP \uparrow	F1 \uparrow (0.5)
ComplEx	53.18 ₍₁₀₎	53.18 ₍₁₀₎	54.39 ₍₁₀₎	52.11 ₍₁₀₎	51.60 ₍₁₀₎	50.97 ₍₉₎	56.31 ₍₈₎	55.30 ₍₉₎	55.33 ₍₈₎
ConvE	49.65 ₍₁₁₎	49.79 ₍₁₁₎	66.38 ₍₈₎	50.03 ₍₁₁₎	50.07 ₍₁₁₎	27.32 ₍₁₁₎	52.43 ₍₁₀₎	53.17 ₍₁₀₎	29.15 ₍₁₀₎
DistMult	53.94 ₍₉₎	53.51 ₍₉₎	53.71 ₍₁₁₎	52.49 ₍₉₎	52.12 ₍₉₎	49.50 ₍₁₀₎	55.29 ₍₉₎	56.26 ₍₈₎	56.51 ₍₇₎
DistMult + ASA	64.85 ₍₈₎	67.29 ₍₈₎	64.30 ₍₉₎	76.45 ₍₆₎	78.63 ₍₆₎	69.48 ₍₆₎	67.86 ₍₇₎	68.72 ₍₆₎	66.47 ₍₅₎
metapath2vec	94.15 ₍₆₎	94.01 ₍₆₎	87.48 ₍₆₎	70.98 ₍₇₎	70.02 ₍₇₎	65.34 ₍₈₎	73.47 ₍₃₎	70.88 ₍₄₎	16.35 ₍₁₁₎
HAN	87.57 ₍₇₎	88.15 ₍₇₎	77.35 ₍₇₎	64.66 ₍₈₎	61.24 ₍₈₎	65.45 ₍₇₎	52.33 ₍₁₁₎	52.06 ₍₁₁₎	41.88 ₍₉₎
GATNE	96.25 ₍₅₎	94.77 ₍₅₎	91.36 ₍₄₎	84.47 ₍₅₎	82.32 ₍₅₎	76.83 ₍₅₎	69.72 ₍₅₎	67.22 ₍₇₎	61.87 ₍₆₎
R-GCN	97.16 ₍₄₎	95.87 ₍₄₎	94.52 ₍₂₎	92.38 ₍₄₎	92.18 ₍₄₎	83.35 ₍₃₎	68.69 ₍₆₎	69.95 ₍₅₎	66.75 ₍₄₎
R-GCN + ASA	98.37 ₍₃₎	97.87 ₍₃₎	94.21 ₍₃₎	93.19 ₍₃₎	93.11 ₍₃₎	79.33 ₍₄₎	69.96 ₍₄₎	73.20 ₍₃₎	67.90 ₍₃₎
RelGNN - ASA	98.84 ₍₂₎	98.55 ₍₂₎	95.17 ₍₁₎	94.39 ₍₂₎	93.41 ₍₂₎	86.32 ₍₁₎	74.94 ₍₂₎	73.91 ₍₂₎	70.45 ₍₂₎
RelGNN	99.14 ₍₁₎	99.01 ₍₁₎	90.44 ₍₅₎	96.44 ₍₁₎	96.08 ₍₁₎	83.89 ₍₂₎	77.68 ₍₁₎	77.78 ₍₁₎	74.43 ₍₁₎

Table 2: Test results on benchmarks and the real-world dataset. AUC denotes the *area under the receiver operating characteristic curve* value and AP denotes the *average precision* corresponding to the *area under the precision-recall curve* value. \uparrow indicates that the higher the score the better the performance. AUC, AP and F1 are reported as percentage. The cutoff threshold for F1 is 0.5. (\dots) after each score indicates the ranking of the method w.r.t. the specific setting. Underlined numbers are quoted from (Cen et al. 2019).

the difference in score distribution between different models. R-GCN models fall short by relatively small margins especially on Amazon and Youtube due to their capability of modeling the heterogeneous neighborhoods with dedicated message encoding functions. However, R-GCN is not as good on Company due to its lack of emphasis on the multi-relational phenomenon and the attribute information in the final entity embedding. GATNE’s performances are consistent across all three datasets. Although GATNE’s final embeddings are edge type aware and are mix of topological and attribute information, they do not provide sufficient expressive power to beat RelGNN partially due to their lack of emphasis on the importance of aggregated information via different edge types. The metapath2vec models come after the previous two baselines. However, it gives better performance on the Company dataset especially in terms of AUC and AP. In sum, RelGNN consistently outperforms the competitors and could benefit from ASA training.

ASA Negative Sampling. Next, we report the results of a set of experiments which compares our proposed ASA negative sampler against other sampling methods. Since Amazon and YouTube are number coded, we focus on the Company dataset from which we can really tell whether or not a sampled relationship is a false negative. We use MRR and Hit@k to evaluate the methods where the perfect scores can be achieved by scoring the true relationships higher than all the respective negative samples.

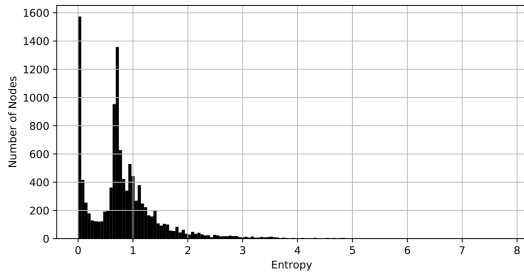
Table 3 reports the performance of RelGNN trained by

different negative sampling techniques. We vary the pool size for NSCaching and ASA (constant $\mu = 0.1$). A negative sample pool is randomly sampled from the complete negative sample set. NSCaching and ASA only evaluate and pick the negative samples for training from the pool. (Zhang et al. 2019b) points out that a smaller pool size to some degree can help the sampler avoid selecting false negatives. The reason is intuitive which is that a smaller pool will likely to have less amount of false negative than a larger one. The NSCaching and the ASA with pool size as 1 would be equivalent to the Random method. As expected, Random sampler gives the worst results as it fails to identify more difficult ones for the model to learn. Even with a very small pool size, ASA₁₀ outperform NSCaching₁₀ in most of the measures. As the pool size grows bigger, NSCaching and ASA are able to explore more negative samples. However, without effective mechanism to avoid false negatives, NSCaching’s performance suffers as it evaluates more negative samples. Most of the measurements go lower as the pool size increases. On the other hand, ASA sampler can explore more negative samples while keeping lower false negative rate. As a result, ASA samples with larger pool size have better performance especially in Hit@k in this case.

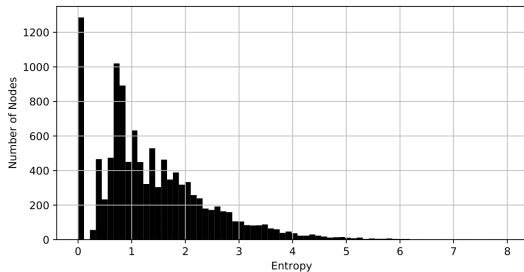
Next, we show how μ can affect the model performance. We measure the Hit@k on Company of RelGNN trained by ASA samplers with different μ . As shown in Figure 2, we observe that there is an increasing trend of Hit@k up until a specific μ value and the measurements decrease as the μ

Methods	Company			
	MRR \uparrow	Hit@1 \uparrow	Hit@10 \uparrow	Hit@30 \uparrow
Random	.0457(7)	2.37(7)	7.71(7)	14.94(7)
NSCaching ₁₀	.0789(2)	5.38 (1)	11.77(4)	19.20(4)
NSCaching ₁₀₀	.0695(5)	3.89(6)	11.50(5)	18.09(5)
NSCaching ₅₀₀	.0658(6)	3.99(4)	10.84(6)	16.75(6)
ASA ₁₀	.0818 (1)	5.18(2)	13.32(3)	22.01(3)
ASA ₁₀₀	.0754(3)	3.95(5)	14.35 (1)	23.71 (1)
ASA ₅₀₀	.0751(4)	4.32(3)	13.71(2)	22.99(2)

Table 3: Test results on *Company*. Hit@k is in percentage.



(a) RelGNN layer 1.



(b) RelGNN layer 2.

Figure 3: Entropy of attention distributions.

further increases. This demonstrates the effectiveness of μ on controlling the hardness of the negative samples used for the training. When the μ is too small, it causes the ASA to select some hard ones which may be false negatives. When the value is too large, most of the negative samples are then trivial cases from which the model may not gain too much. Figure 2 also shows that a good μ (0.15) value to have negative samples with a right level of hardness is relatively easy to find during hyperparameter tuning.

4.2 Ablation Study

We also conduct experiments to demonstrate the effect of each designed component on the model performance.

Attention Mechanism. The RelGNN network without attention mechanism is equivalent to R-GCN. To study the contribution of attention in learning, we use RelGNN-ASA,

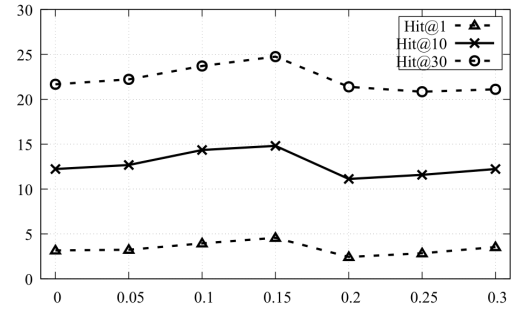


Figure 2: Hit@k by varying μ .

which is a variant of RelGNN trained with a random negative sampler, and compare it against R-GCN on *Company* dataset. As shown in Table 2, RelGNN-ASA outperforms R-GCN across all datasets in every measure proving that the attentions are effective. For further analysis, we measure the entropy of attention distribution over nodes. Figure 3(a,b) shows the node-wise attention entropy distribution learned by two RelGNN layers. Note that we use a single head attention setting in this experiment. The plots have a distribution of lower entropy which means high degrees of concentration. Low entropy attention distributions provide more selective message passing capability than simple pooling operations such as mean pooling. It shows that our model RelGNN is able to handle complex neighborhood by learning a few neighbors’ messages which are more important than others. This validates our claim that not all neighbors are important in a multi-relational graph.

ASA Negative Sampling. Next, we test the effect of ASA by replacing it with a random negative sampler. This refers to the comparison between RelGNN and RelGNN-ASA in Table 2. RelGNN outperforms RelGNN-ASA in AUC and AP. It sometimes has lower F1 because the models are actually only better with a lower cutoff threshold. With the hard negative samples identified by ASA in this case, RelGNN models act “cautiously” with the score distribution moving towards the lower end of the scale. We also extend this test to other representative models from each category which refers to the comparisons between R-GCN and R-GCN+ASA, and DistMult and DistMult+ASA. The results in Table 2 demonstrates the effectiveness of ASA component.

5 Conclusion

In this paper, we describe an end-to-end solution for relationship prediction in heterogeneous graphs. Particularly, we propose a message passing-based graph attention model – RelGNN to exploit the heterogeneity of the neighborhood by leveraging the edge semantics. We also introduce an effective hard negative sampling method called ASA which reduces the false negative rate by consulting the positive relationships. We demonstrate the strength of RelGNN together with ASA by comparing against state-of-the-art methods on relationship prediction across established benchmarks and a real industrial dataset.

Ethical Impact

The views and conclusions expressed in this section are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the affiliated corporation.

The results in this paper are a set of empirically verified techniques which can improve the expressive power of the graph neural networks particularly for processing multi-relational, heterogeneous graphs. In this work, we propose two contributions – one towards information propagation via attention, second for model training to achieve better generalization. Our proposed model can leverage edge semantics to determine the importance of information propagated from neighbors to improve the quality of the node embedding. For optimization, we propose the adaptive self-adversarial method which could generate hard negative samples while reducing the false negative rate. We specifically deal with the relationship prediction task which is of particular interest to many application areas ranging from social media, user recommendations, network analysis, and scientific discovery. Our work was initiated with the intention to enable and/or advance capabilities and/or services for our partners and clients whose businesses and practices are legislated by respective laws and policies. The potential applications include fraud detection for finance institutions, product recommendation for large enterprises etc.

While aiming for positive impact such as saving costs and providing better customer services and experiences, and deriving business insights, we do acknowledge the possibility of negative effects especially when our work is made available to the general public. For example, our proposed techniques could assist scammers to identify more vulnerable subjects in a network of people. Enterprises who commit to not use sensitive data such as certain demographic information for business purposes may indirectly make use of those information derived from the insensitive ones by our algorithm. The other application scenario is online social networks, the possible negative impact is privacy concerns. Our method may derive information which is otherwise unavailable or hidden such as predicting likes/dislikes of a user, deriving insights on non-obvious relationships between users. However, to the best of our knowledge, our work does not create a new line of threats to the society and the above examples which are open questions that remain true to many other works as well. While it is out of scope of this paper to address such issues, we appreciate and agree with the AAAI 2021 organizing committee for bringing attention to the topic of impact beyond technical contributions.

References

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V. F.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; Gülçehre, Ç.; Song, H. F.; Ballard, A. J.; Gilmer, J.; Dahl, G. E.; Vaswani, A.; Allen, K. R.; Nash, C.; Langston, V.; Dyer, C.; Heess, N.; Wierstra, D.; Kohli, P.; Botvinick, M.; Vinyals, O.; Li, Y.; and Pascanu, R. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR* abs/1806.01261.

Cai, L.; and Wang, W. Y. 2018. 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*, 1470–1480. Association for Computational Linguistics.

Cen, Y.; Zou, X.; Zhang, J.; Yang, H.; Zhou, J.; and Tang, J. 2019. Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1358–1368.

Chen, Y.; Wu, L.; and Zaki, M. 2020a. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings. *Advances in Neural Information Processing Systems* 33.

Chen, Y.; Wu, L.; and Zaki, M. J. 2020b. Reinforcement learning based graph-to-sequence model for natural question generation. *International Conference on Learning Representation*.

Chen, Y.; Wu, L.; and Zaki, M. J. 2020c. Toward Subgraph Guided Knowledge Graph Question Generation with Graph Neural Networks. *arXiv preprint arXiv:2004.06015*.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2D Knowledge Graph Embeddings. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 1811–1818. AAAI Press.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. meta-path2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 135–144. ACM.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, 1024–1034.

Hayashi, K.; and Shimbo, M. 2017. On the Equivalence of Holographic and Complex Embeddings for Link Prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 554–559.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, 2704–2710.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2020. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *CoRR* abs/2002.00388. URL <https://arxiv.org/abs/2002.00388>.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations*.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-Paced Learning for Latent Variable Models. In *Advances in Neural Information Processing Systems*, 1189–1197.

Li, J.; Tao, C.; Wu, W.; Feng, Y.; Zhao, D.; and Yan, R. 2019. Sampling Matters! An Empirical Study of Negative

- Sampling Strategies for Learning of Matching Models in Retrieval-based Dialogue Systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 1291–1296. Association for Computational Linguistics.
- Mikolov, T.; Kombrink, S.; Burget, L.; Cernocký, J.; and Khudanpur, S. 2011. Extensions of recurrent neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 5528–5531. IEEE.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26: 3111–3119.
- Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, 593–607. Springer.
- Sun, Z.; Deng, Z.; Nie, J.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *7th International Conference on Learning Representations*. OpenReview.net.
- Wang, P.; Li, S.; and Pan, R. 2018. Incorporating GAN for Negative Sampling in Knowledge Representation Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2005–2012. AAAI Press.
- Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference, 2022–2032*.
- Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2019. A Comprehensive Survey on Graph Neural Networks. *CoRR* abs/1901.00596. URL <http://arxiv.org/abs/1901.00596>.
- Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *7th International Conference on Learning Representations*.
- Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations*.
- Yang, C.; Xiao, Y.; Zhang, Y.; Sun, Y.; and Han, J. 2020a. Heterogeneous Network Representation Learning: Survey, Benchmark, Evaluation, and Beyond. *CoRR* abs/2004.00216. URL <https://arxiv.org/abs/2004.00216>.
- Yang, Z.; Ding, M.; Zhou, C.; Yang, H.; Zhou, J.; and Tang, J. 2020b. Understanding Negative Sampling in Graph Representation Learning. *CoRR* abs/2005.09863.
- Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 974–983.
- Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019a. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 793–803.
- Zhang, Y.; Yao, Q.; Shao, Y.; and Chen, L. 2019b. NSCaching: simple and efficient negative sampling for knowledge graph embedding. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 614–625. IEEE.