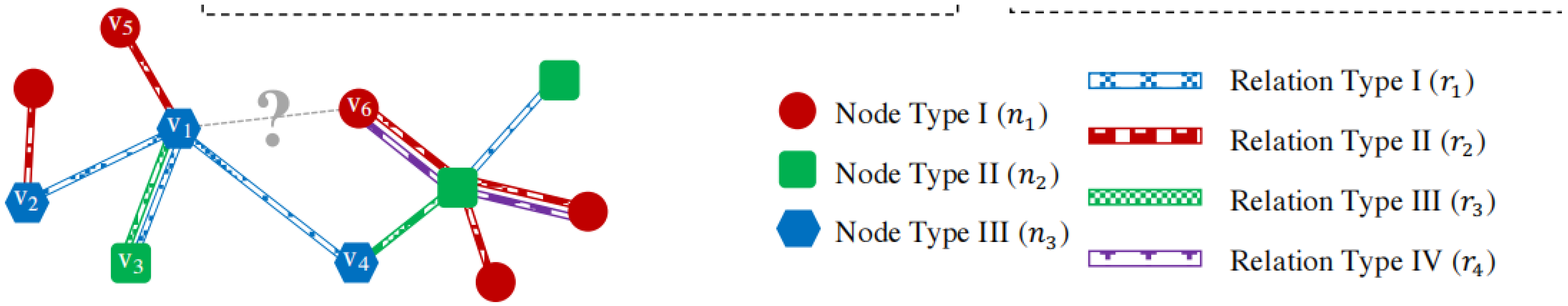
Relation-aware Graph Attention Model With Adaptive Self-adversarial Training

Xiao Qin Nasrullah Sheikh Berthold Reinwald Lingfei Wu

AAAI 2021

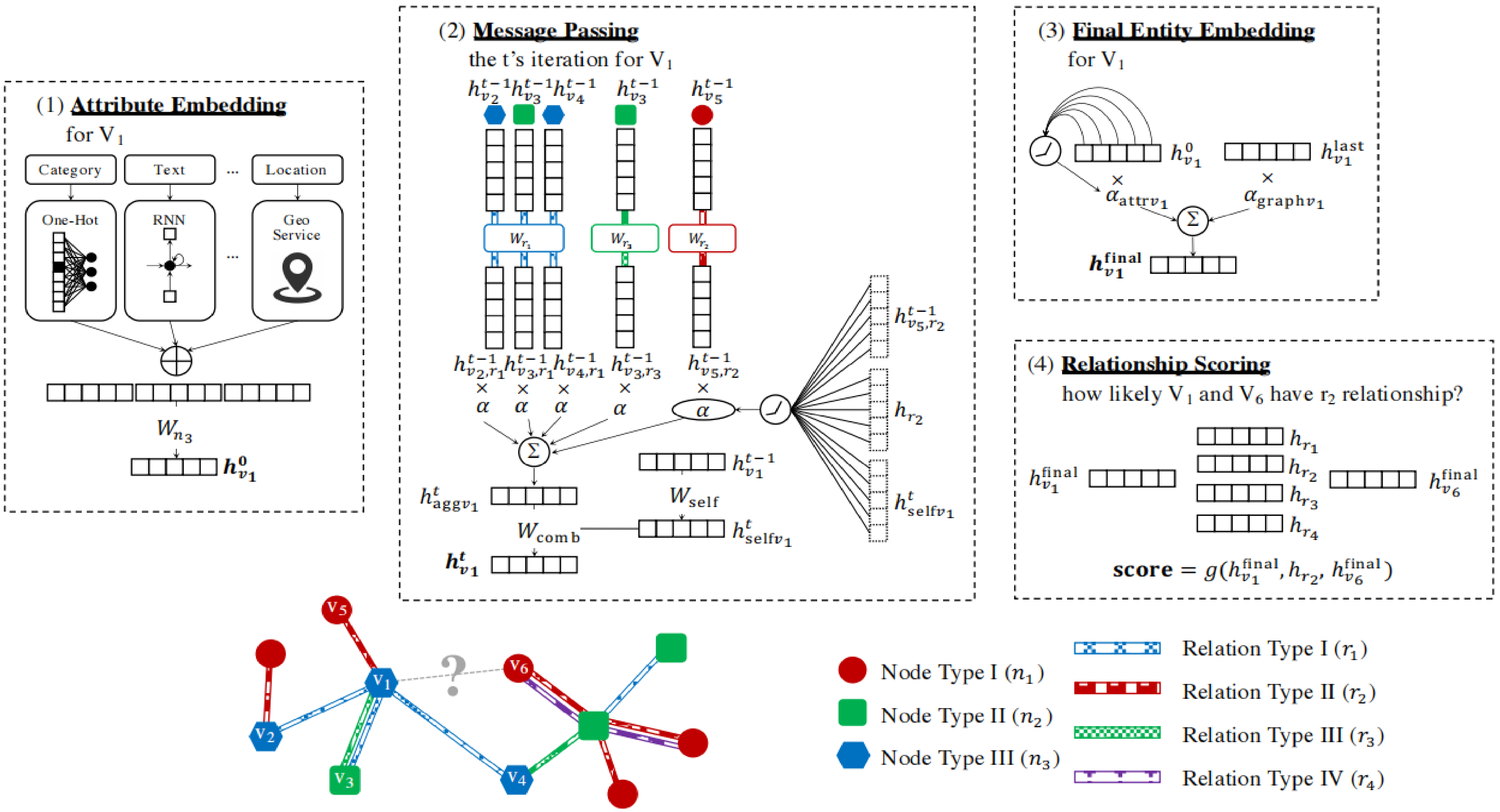
Introduction

heterogeneous graphs on the task of relationship prediction



However, existing research tends to ignore the semantics of the edges, that is the edge information is only used either for graph traversal and/or selection of encoding functions

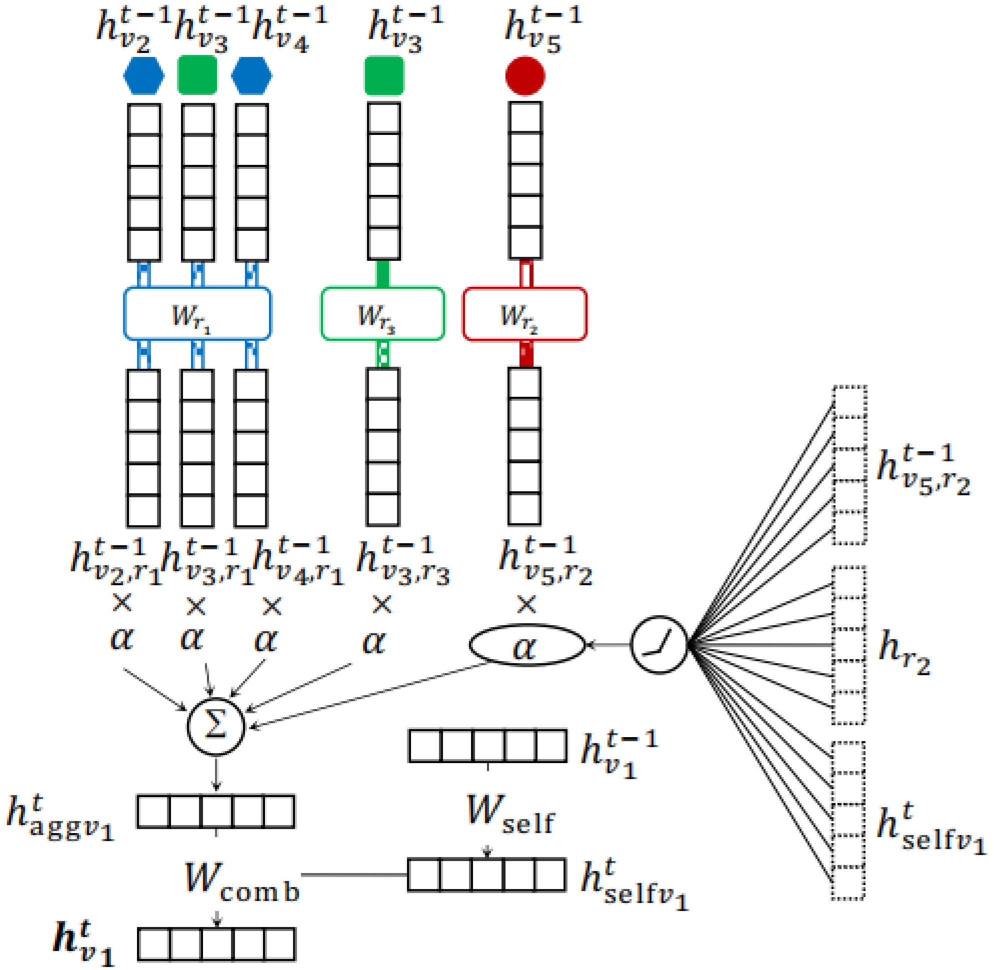
Framework



(1) Attribute Embedding
for V1
Category
Text
Location
One-Hot
RNN
Geo Service
W_{n_3}
$h^0_{v_1}$
(2) Message Passing
the t's iteration for V1
$h^{t-1}_{v_2}$ $h^{t-1}_{v_3}$ $h^{t-1}_{v_4}$ $h^{t-1}_{v_3}$ $h^{t-1}_{v_5}$
W_{r_1} W_{r_3} W_{r_2}
$h^{t-1}_{v_2,r_1}$ $h^{t-1}_{v_3,r_1}$ $h^{t-1}_{v_4,r_1}$ $h^{t-1}_{v_3,r_3}$ $h^{t-1}_{v_5,r_2}$
α
$h^{t-1}_{v_5,r_2}$
h_{r_2}
$h^t_{selfv_1}$
$h^{t-1}_{v_1}$
$h^t_{aggv_1}$
W_{self}
W_{comb}
$h^t_{selfv_1}$
$h^t_{v_1}$
(3) Final Entity Embedding
for V1
$h^0_{v_1}$
$h^{last}_{v_1}$
α_{attrv_1}
α_{graphv_1}
$h^{final}_{v_1}$
(4) Relationship Scoring
how likely V1 and V6 have r2 relationship?
$h^{final}_{v_1}$
h_{r_1} h_{r_2} h_{r_3} h_{r_4}
$h^{final}_{v_6}$
score $= g(h^{final}_{v_1}, h_{r_2}, h^{final}_{v_6})$
V5 V1 V2 V3 V4 V6
?
Node Type I (n_1)
Node Type II (n_2)
Node Type III (n_3)
Relation Type I (r_1)
Relation Type II (r_2)
Relation Type III (r_3)
Relation Type IV (r_4)

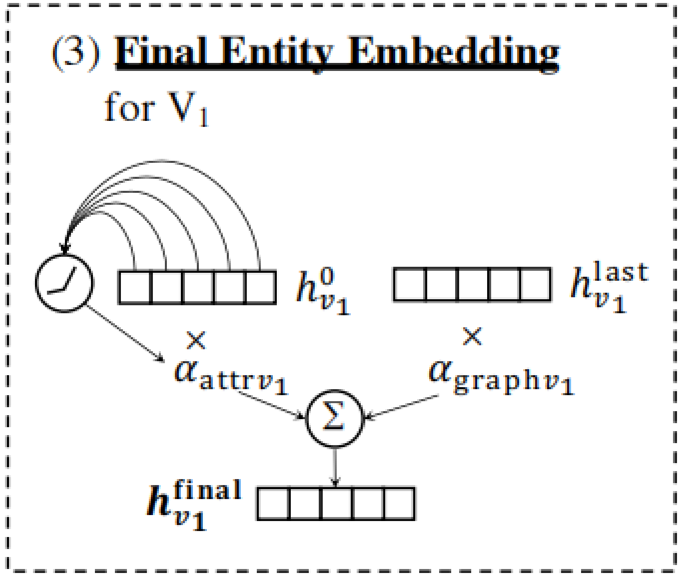
(2) **Message Passing**

the t's iteration for V_1



$$h_{v_i}^{(t)} = \sigma\Big(\frac{1}{L}\sum_{l=1}^{L}\Big(\sum_{r\in R}\sum_{v_j\in N_{v_i}^{r}}\alpha^{l}_{(v_i,v_j)}W_r^{(t-1)}h_{v_j}^{(t-1)} + W_{self}^{(t-1)}h_{v_i}^{(t-1)}\Big)\Big),$$

$$\alpha_{(v_i,v_j)} = \frac{\exp\left(\sigma\left(\boldsymbol{a_e}^{\top}[W_{self}^{(t-1)}h_{v_i}^{(t-1)} \parallel h_r \parallel W_r^{(t-1)}h_{v_j}^{(t-1)}]\right)\right)}{\sum_{r'\in R}\sum_{v_n\in N_{v_i}^{r'}}\exp\left(\sigma\left(\boldsymbol{a_e}^{\top}[W_{self}^{(t-1)}h_{v_i}^{(t-1)} \parallel h_{r'} \parallel W_{r'}^{(t-1)}h_{v_n}^{(t-1)}]\right)\right)} \tag{4}$$



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

$$\alpha_{attr} = \frac{\exp(\sigma(\boldsymbol{a_s}^\top h_{v_i}^{(0)}))}{\exp(\sigma(\boldsymbol{a_s}^\top h_{v_i}^{(0)})) + \exp(\sigma(\boldsymbol{a_s}^\top h_{v_i}^{last}))}, \tag{7}$$

Negative Sampling.

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

only a small subset of all possible negative samples are useful for training

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

$$\downarrow$$

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

Experiment

Methods	Amazon AUC↑	Amazon AP↑	Amazon F1↑(0.5)	Youtube AUC↑	Youtube AP↑	Youtube F1↑(0.5)	Company AUC↑	Company AP↑	Company F1↑(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. ↑ 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. $(\cdots)$ after each score indicates the ranking of the method w.r.t. the specific setting. Underlined numbers are quoted from (Cen et al. 2019).

Sample Result

Methods	Company			
	MRR↑	Hit@1↑	Hit@10↑	Hit@30↑
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.

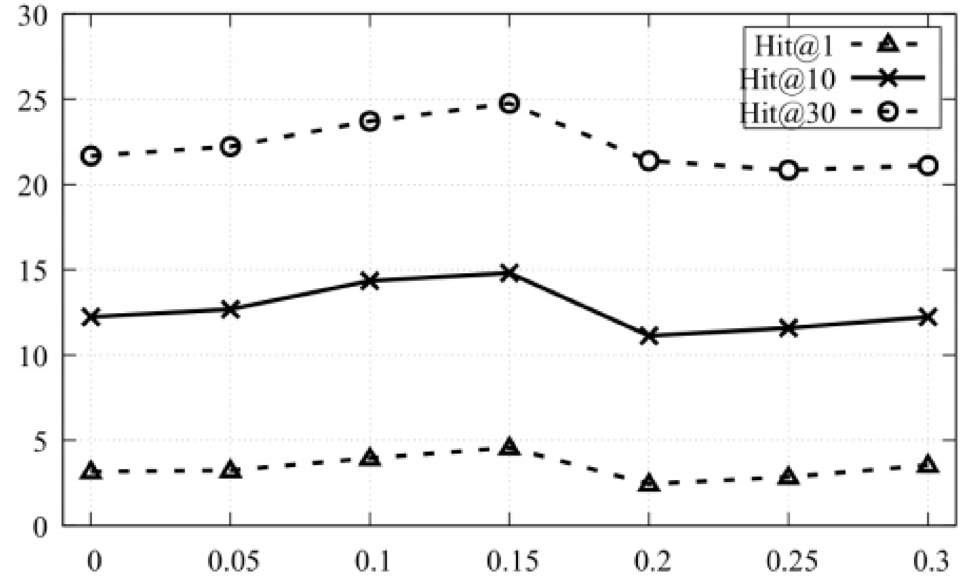
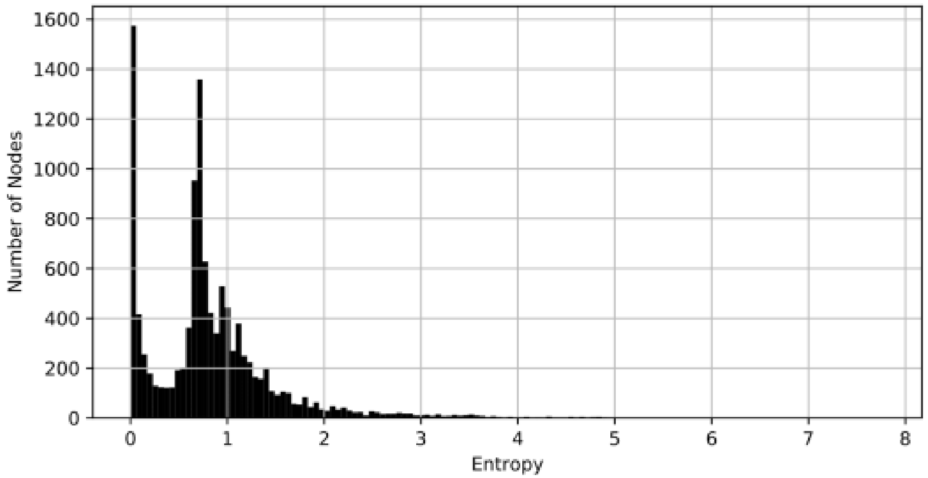
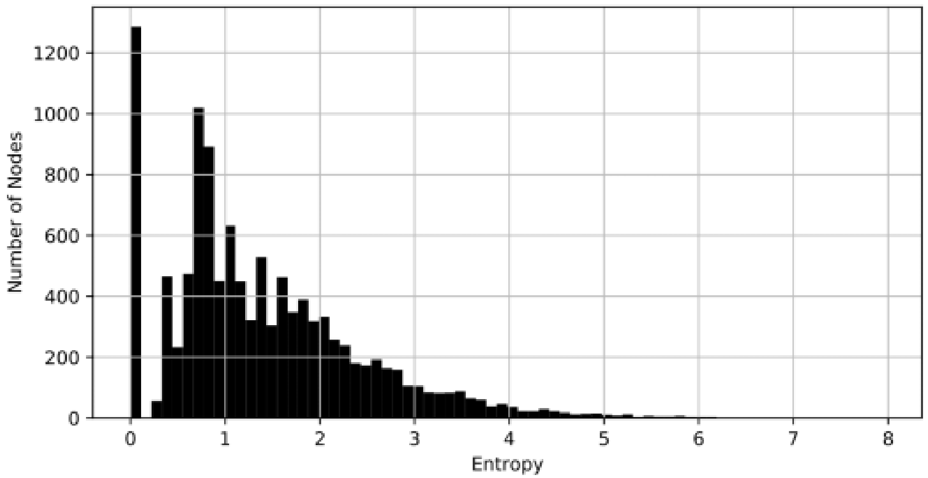


Figure 2: Hit@k by varying μ.



(a) RelGNN layer 1.



(b) RelGNN layer 2.

Figure 3: Entropy of attention distributions.