Recursive Rule Mining in Knowledge Graphs

Aditya Jain, Manish Reddy, Nilay Pochhi

*(Thanks to Vivian for guiding us)

Agenda

- Introduction
- Base Work
- Our contribution
- Datasets
- Results
- Future Work

Introduction

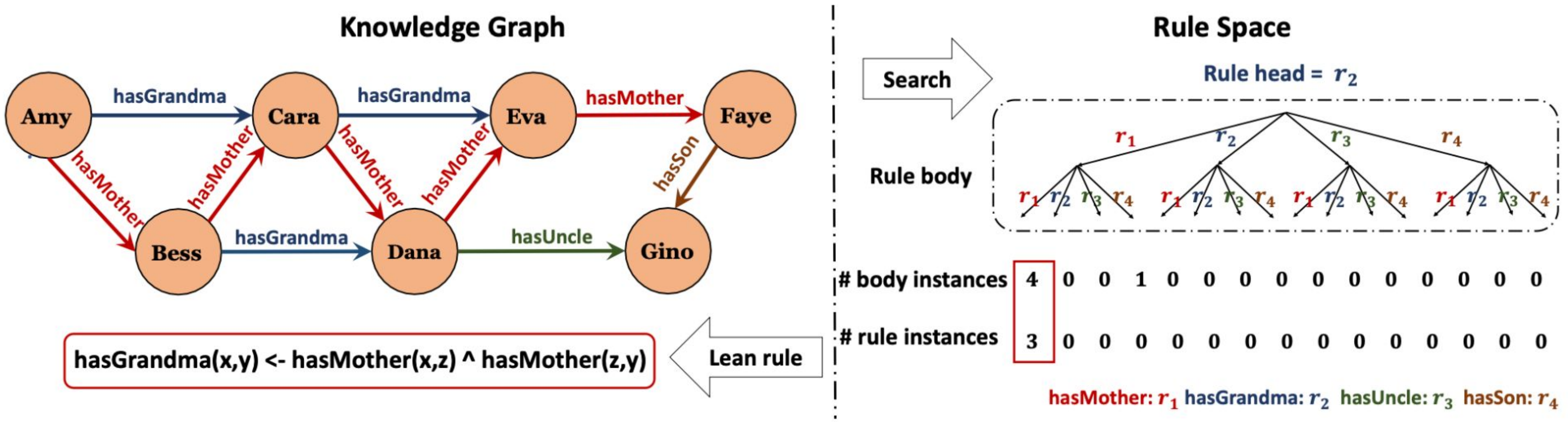
- Logical rule mining is the process to deduce logical rules from the given data
- Logical Reasoning leverages those logical rules to derive missing knowledge and allows easy generalization to unobserved objects

$$\delta_1 := hasGrandma(x, y) \leftarrow hasMother(x, z) \wedge hasMother(z, y)$$

$$\delta_2 := hasUncle(x, y) \leftarrow hasMother(x, z_1) \wedge hasMother(z_1, z_2) \wedge hasSon(z_2, y)$$

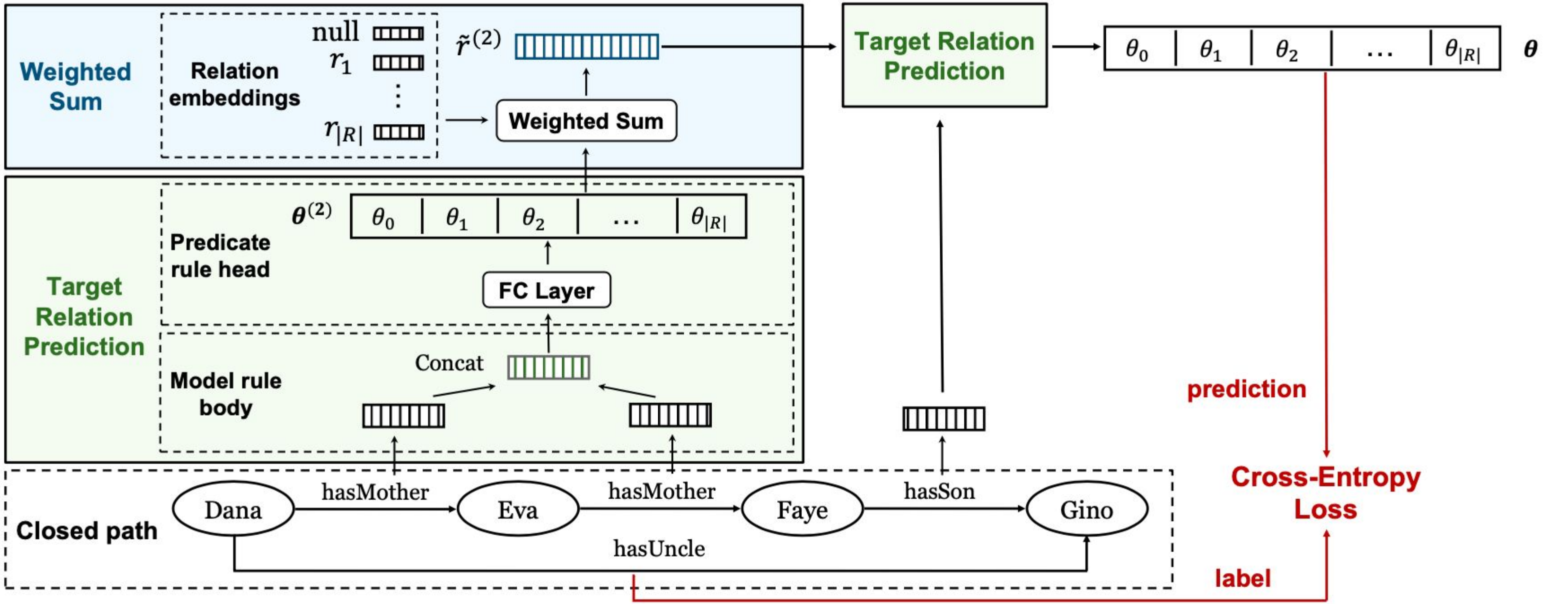
RLogic (Base Work)

Rules sampling in KG:

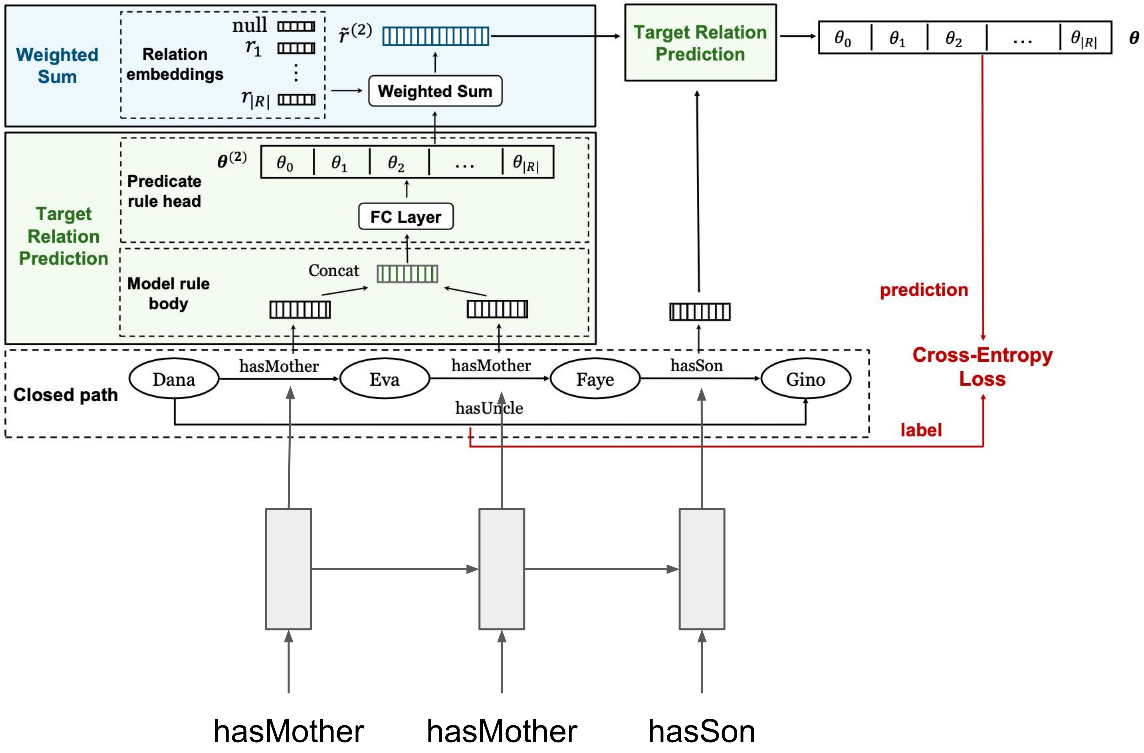


RLogic (Base Work)

Model:



RLogic (Base Work)



Total Loss

=

LSTM Loss

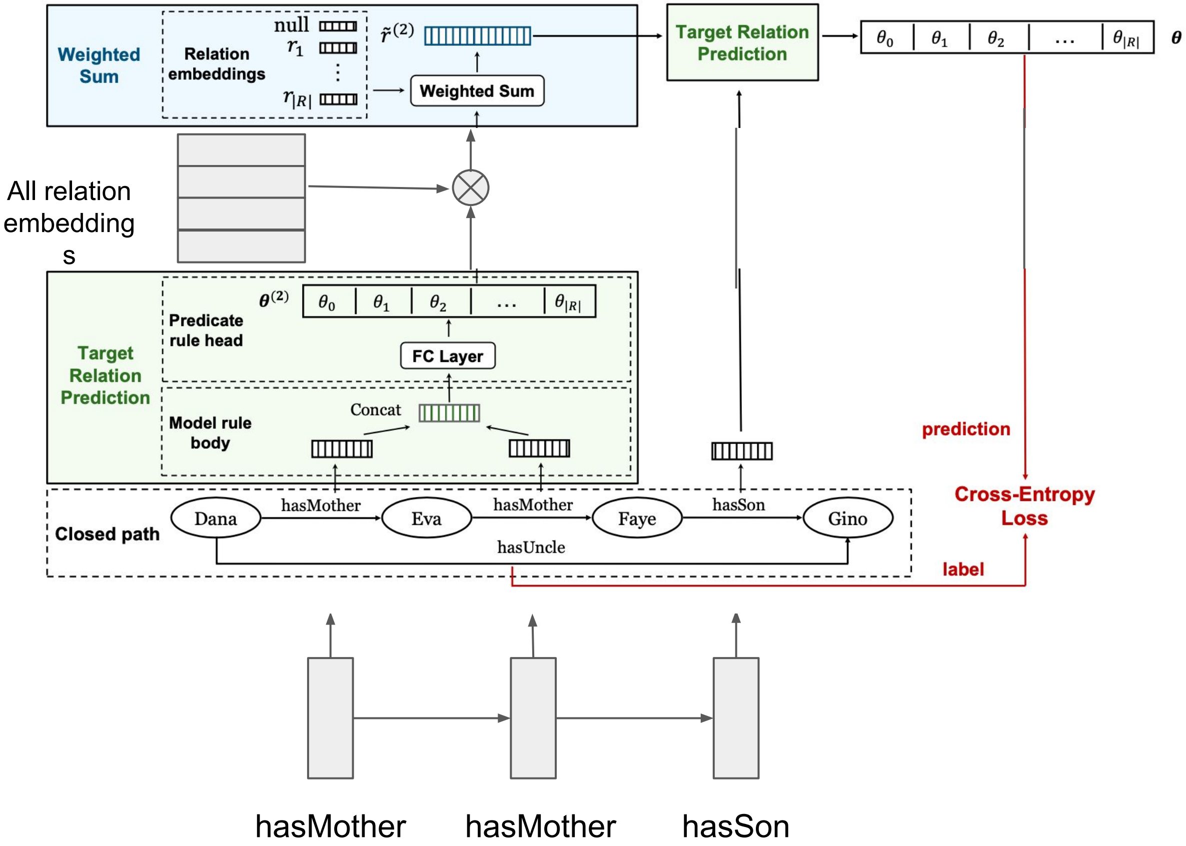
+

Recursive Loss

Proposed Methods

- RLogic with naïve attention
- RLogic with advanced attention
- RLogic with GCN

RLogic with naïve attention



Total Loss

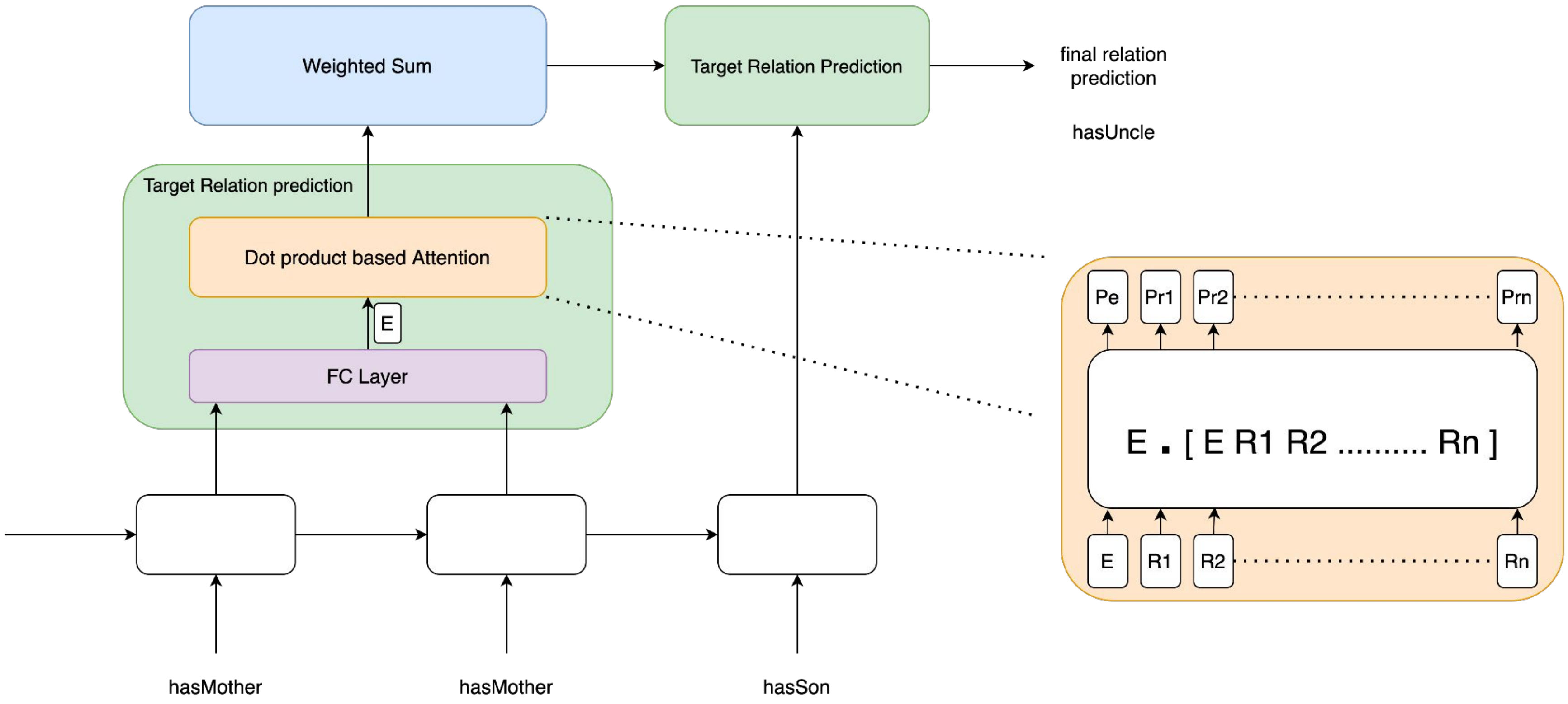
=

LSTM Loss

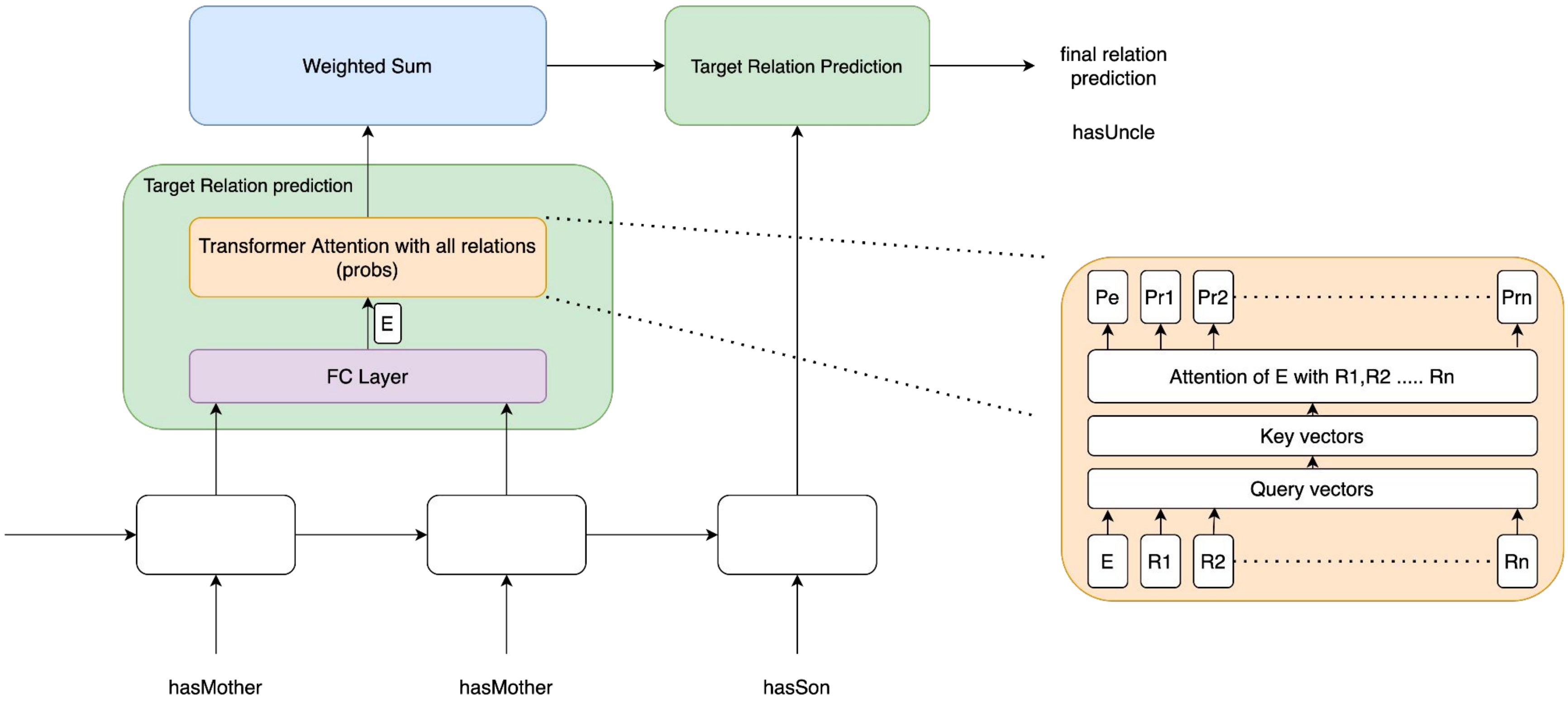
+

Recursive Loss

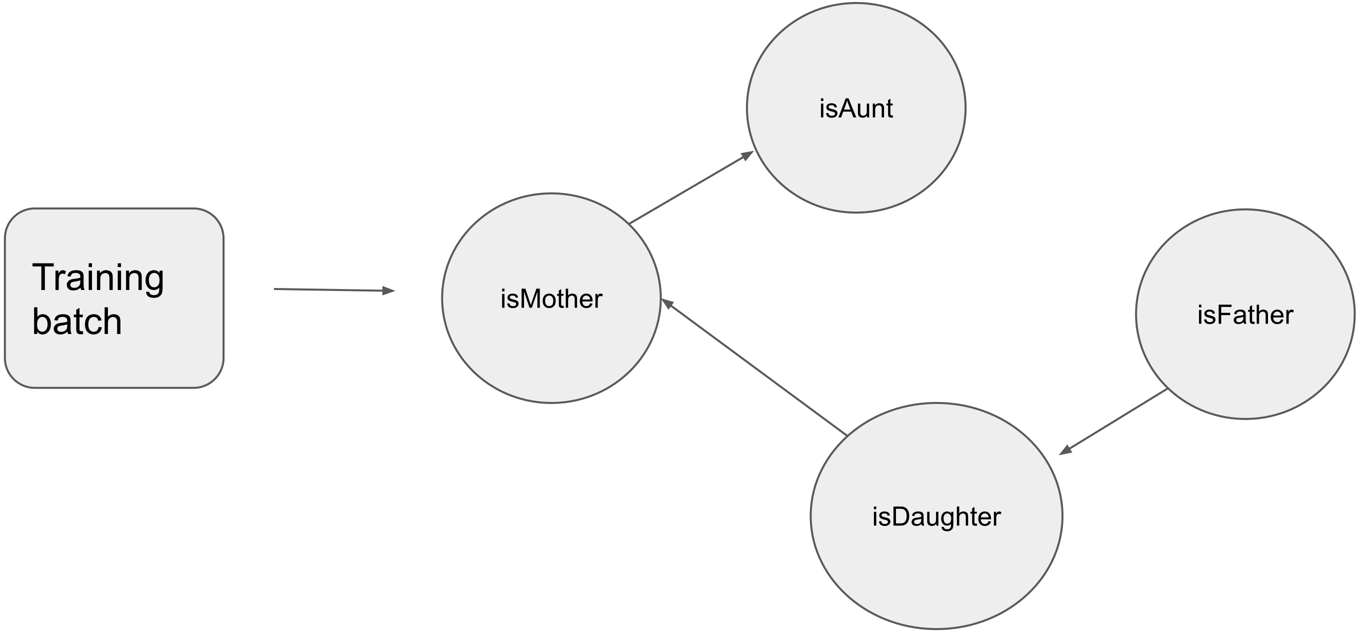
RLogic with naïve attention



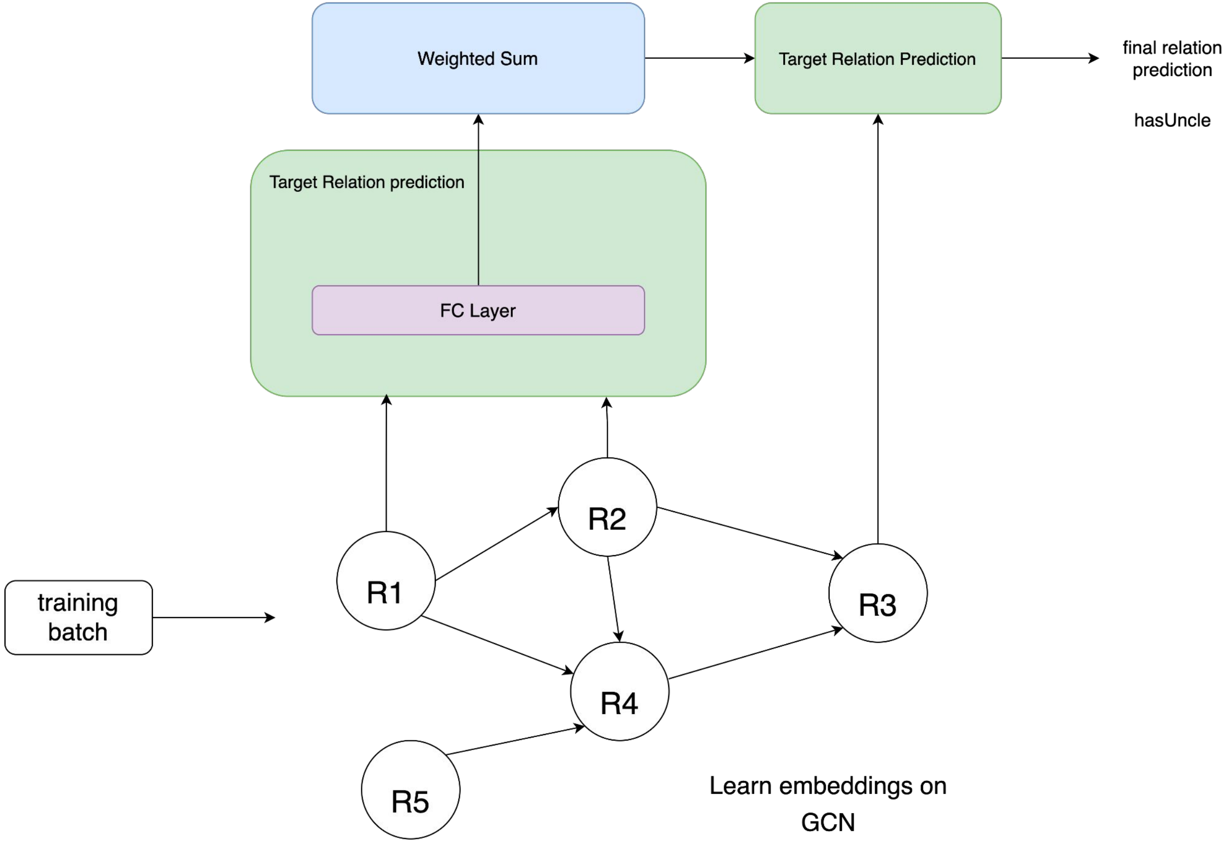
RLogic with advanced attention



RLogic with GCN



RLogic with GCN



Datasets

- Family
  - 3007 entities
  - 12 relations
- FB15-237
  - 14541 entities
  - 237 relations
- WN-18RR
  - 40943 entities
  - 11 relations

Metrics Used

- Mean Reciprocal Rank

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\mathrm{rank}_i}$$

- Hits@10

Hits@L = $$\frac{|U_{hit}^{L}|}{|U_{all}|}$$

Results

Family Dataset

RLogic original top 10 rules	
0.998	husband <-- daughter, father
0.997	daughter <-- husband, daughter
0.997	daughter <-- inv_husband, daughter
0.996	aunt <-- mother, sister
0.996	daughter <-- daughter, sister
0.996	daughter <-- inv_mother, sister
0.996	uncle <-- mother, brother
0.995	brother <-- wife, inv_wife
0.995	husband <-- inv_mother, father
0.994	mother <-- inv_son, inv_husband

RLogic with transformer attention top 10 rules	
0.997	brother <-- brother, brother
0.996	brother <-- inv_sister, brother
0.994	brother <-- mother, son
0.994	father <-- brother, father
0.994	aunt <-- sister, aunt
0.993	father <-- sister, father
0.993	father <-- inv_brother, father
0.993	son <-- husband, son
0.993	wife <-- inv_father, mother
0.991	aunt <-- inv_sister, aunt

Results

Family Dataset

RLogic original top 10 rules (len 3)	
0.999	daughter <-- inv_father, inv_brother, sister
0.999	daughter <-- inv_father, sister, sister
0.999	niece <-- brother, inv_husband, daughter
0.999	daughter <-- inv_father, inv_sister, sister
0.999	niece <-- sister, inv_husband, daughter
0.999	daughter <-- inv_father, brother, sister
0.999	brother <-- inv_son, inv_mother, brother
0.999	brother <-- mother, inv_mother, brother
0.999	niece <-- sister, inv_wife, daughter
0.998	niece <-- inv_brother, inv_sister, daughter

RLogic with transformer attention top 10 rules (len 3)	
0.993	aunt <-- inv_daughter, niece, aunt
0.993	daughter <-- daughter, inv_daughter, daughter
0.992	sister <-- inv_aunt, father, sister
0.992	sister <-- inv_sister, inv_brother, sister
0.991	aunt <-- father, niece, aunt
0.991	sister <-- inv_sister, inv_sister, sister
0.991	sister <-- inv_sister, brother, sister
0.991	aunt <-- inv_daughter, inv_uncle, aunt
0.991	nephew <-- inv_uncle, mother, son
0.991	aunt <-- inv_son, niece, aunt

Results

Family Dataset

Model	Rule Length	MRR	Hit@10
RLogic (baseline)	2	0.857	0.9527
	3	0.878	0.9690
RLogic (with naive attention)	2	0.858 ≈	0.9527 ≈
RLogic (with advanced attention)	2	0.856 ≈	0.9530 ≈
	3	0.839 ≈	0.9314 ≈
RLogic (with GCN)	2	0.475 ↓	0.602 ↓

Combine learned rules with KG embedding for KG completion task

Results

FB15K-237 Dataset

Rule len: 2

Model	MRR	Hit@10
RLogic (baseline)	0.056	0.079
RLogic (with naive attention)	0.0536 ≈	0.140 ↑
RLogic (with advanced attention)	0.146 ↑	0.356 ↑
RLogic (with GCN)	-	-

- Evaluated on 100 test cases with top 10000 (1%) out of 10M predicted rules
- GCN method requires longer training times and computation power on fb15k-237 dataset

Results

WN-18RR Dataset

Rule len : 2

Model	MRR	Hit@10
RLogic (baseline)	0.0563	0.109
RLogic (with naive attention)	0.0507	0.107
RLogic (with advanced attention)	0.0465	0.097
RLogic (with GCN)	-	-

- Evaluated on 1000 test cases with top 500(10%) out of 5K predicted rules
- Adding more rules to the evaluation will proportionally increase the MRR and Hit@10
- GCN method requires longer training times and computation power on wn-18rr dataset

Future work

- Finetune GCN and combine GCN with attention based improvements
- Try the attention based methods with different sequence lengths
- Train for more epochs and evaluate with full set of predicted rules
- Usage of BERT based techniques for encoding the text of a relation