Sparsity
in
Knowledge Graph Completion

Yang Yu

Preliminary

Knowledge Graph Completion Task

Goal: Predicting Knowledge Instances

e.g., mining missing entities, relationships, or discovering new facts

(Beijing, capitalOf, ?) / (?, capitalOf, China), (Beijing, ?, China)

or confi = (Beijing, capitalOf, China)

$\underbrace{\text{(Entity, Predicate,}}_{\text{Query}}\ \underbrace{\text{?}}_{\text{Answer}}\text{)}$

Preliminary

Main-stream Solutions

- **Embedding-based**

  Build Semantic Space for Both Entity and Relation

  by Individual Fact

  [1] Translating embeddings for modeling multi-relational data. NIPS. 2013 [2] Factorizing YAGO: scalable machine learning for linked data. YAGO. 2012

- **Path-based**

  Build Relation Feature

  by Completed Path between Entities

  [1] Random walk inference and learning in a large scale knowledge base. EMNLP. 2011 [2] Efficient and expressive knowledge base completion using subgraph feature extraction. EMNLP. 2015.

- **Rule-based**

  Generate rules

  by Statistics

  [1] Robust Discovery of Positive and Negative Rules in Knowledge Bases. ICDE. 2018 [2] Fast and Exact Rule Mining with AMIE 3. ESWC. 2020

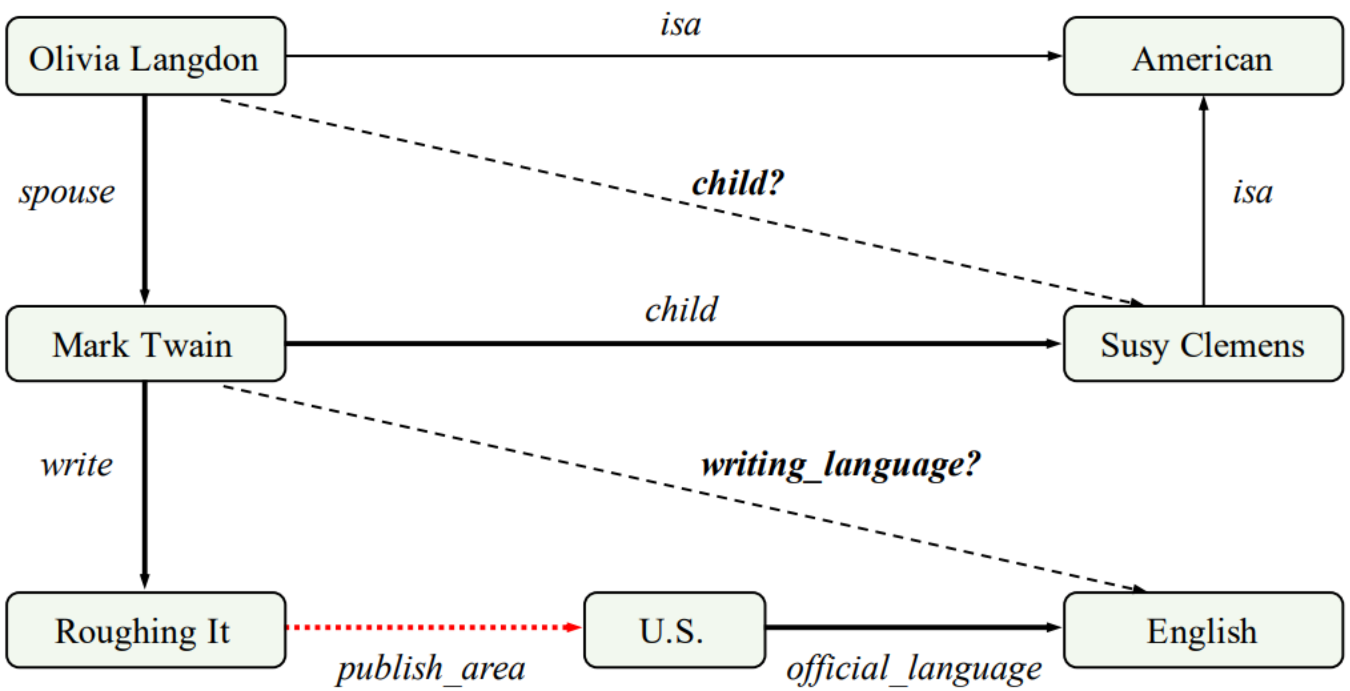
Preliminary

Sparsity

Imbalance data and Insufficient information

Incompleteness

Lack the important paths



Preliminary

View of Data

Sparsity:

stable

sparse

Noise

Simple is better

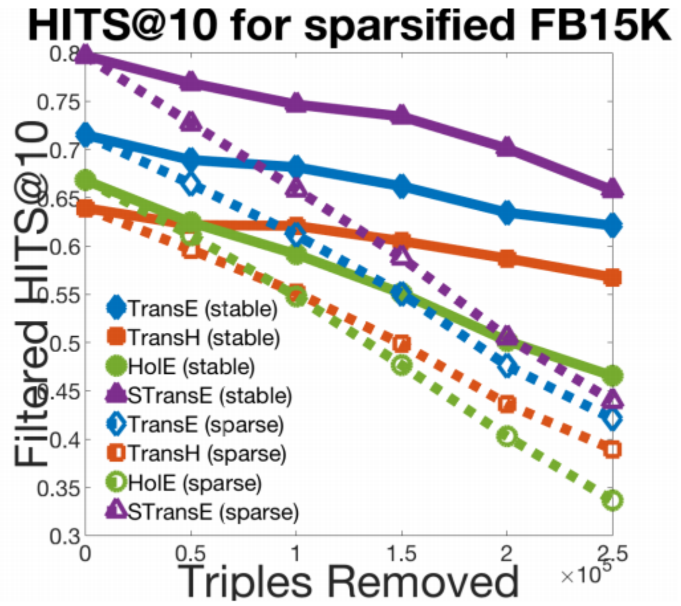


Figure 1: Triples are removed from FB15K to preserve relational density (`stable`, solid) or to increase sparsity (`sparse`, dotted). Sparse training sets have a pronounced impact on the learned embedding, as measured by HITS@10 on the test set.

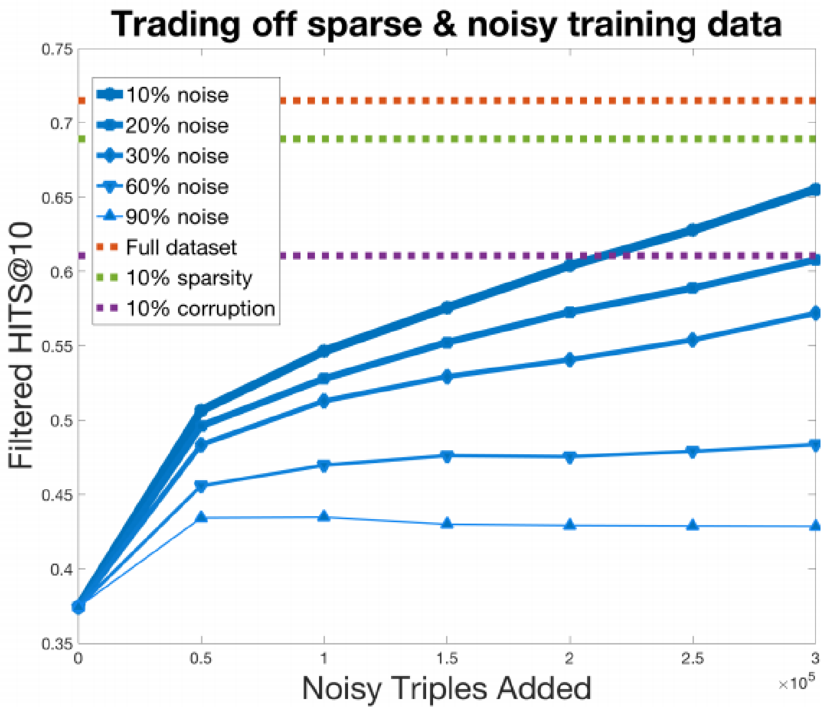


Figure 3: Starting with a sparse training set, adding unreliable triples can help embedding performance recover if the noise level is low.

Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short. EMNLP. 2017

Preliminary

How to measure sparsity ?

$$Relation\_Density = \frac{\|T\|}{\|R\|}, Entity\_Density = \frac{2\|T\|}{\|E\|}$$ [1]

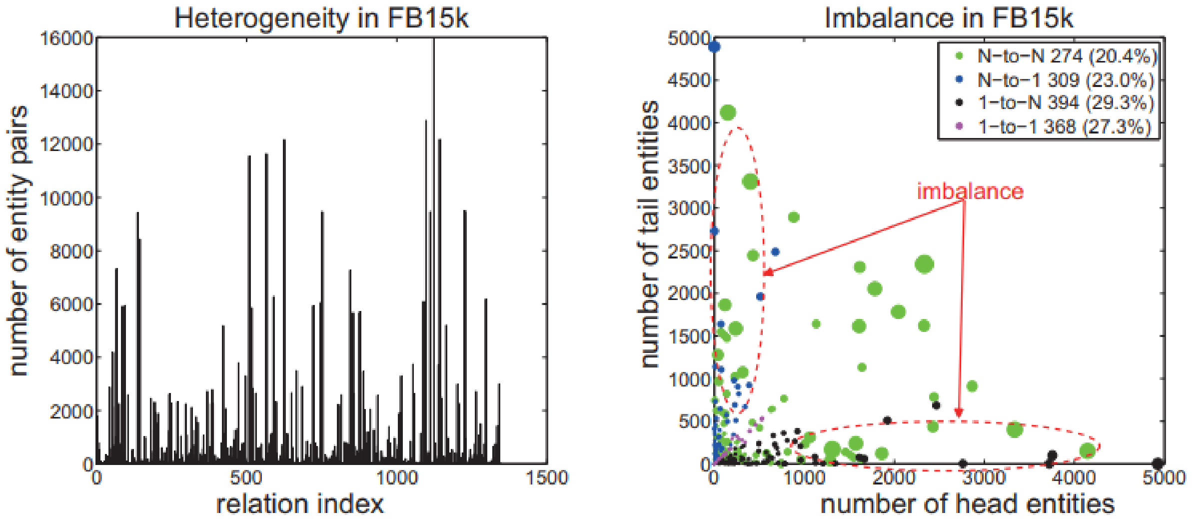
$$\text{Sparsity(e)} = 1 - \frac{freq(e) - freq_{min}}{freq_{max} - freq_{min}}$$ [2]

[1] Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short. EMNLP. 2017

[2] Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning. WWW. 2019

TranSparse

- Use Sparse Matrix (**Heterogenous**)
- Two separate Matrix for head and tail entity (**Unbalance**)



(a) Heterogeneity (b) Imbalance

Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. AAAI. 2016

TranSparse

Share

set a sparse transfer matrix $M_r(\theta_r)$ and a translation vector r for each relation r (TransR+)

$$h_p = M_r(\theta_r)h, \;\; t_p = M_r(\theta_r)t$$

where

$$\theta_r = 1 - (1 - \theta_{\mathbf{min}})N_r / N_{r^\star}$$

N_r is the number of entity pairs about relation,

$N_{r^\star}$ is the maximum number of N_r

$\theta_{\mathbf{min}}$ is a hyper-parameter

Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. AAAI. 2016

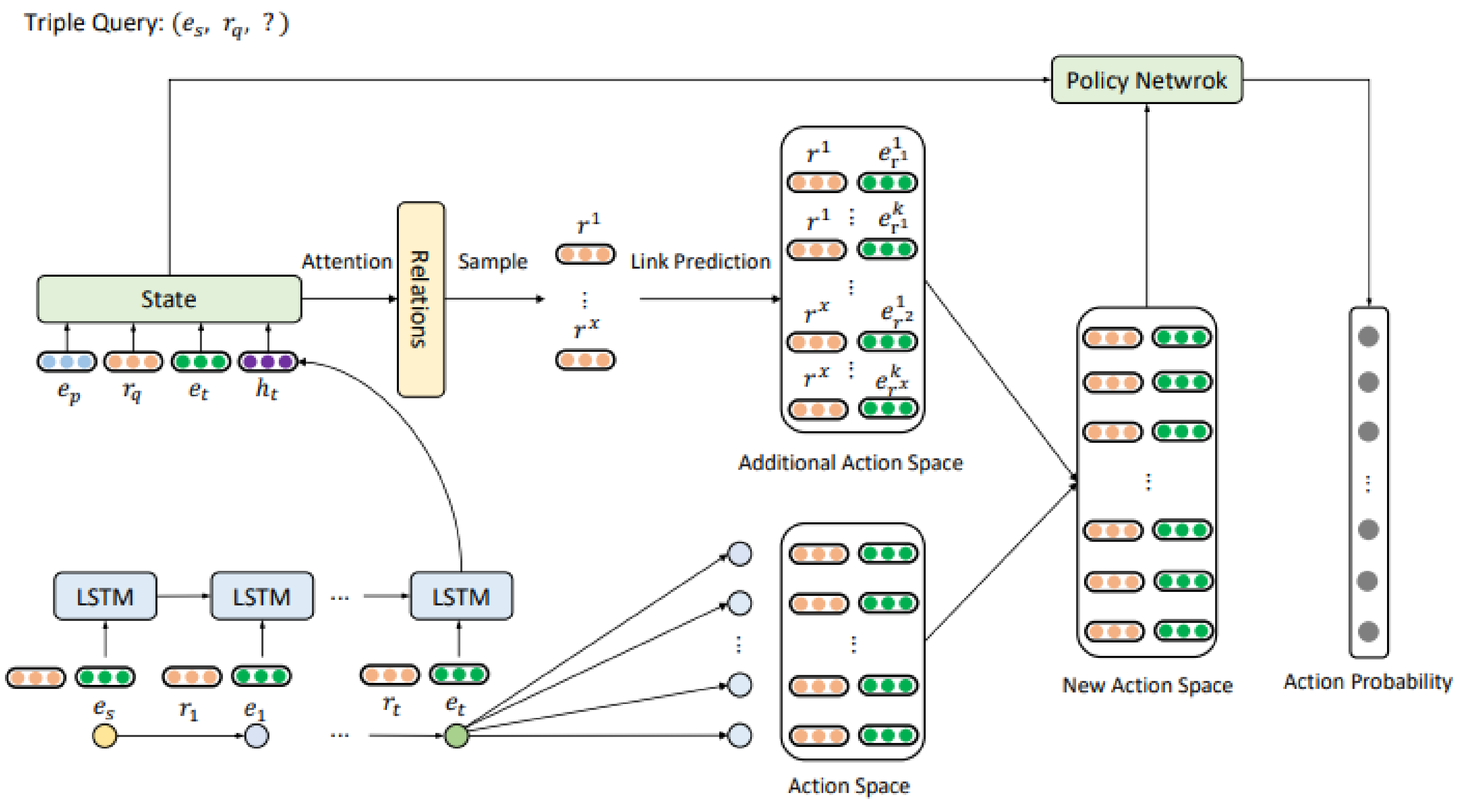
DacKGR

Anticipation Strategy

Inject the pre-trained model as anticipation information

Dynamic Completion

dynamically adds some additional relations



Dynamic Anticipation and Completion for Multi-Hop Reasoning over Sparse Knowledge Graph. EMNLP. 2020

DacKGR

Difference from Previous RL-based Works

Dynamic Anticipation

New state representation

$$s_t = [r_q; e_t; h_t] \Rightarrow s_t = [e_p; r_q; e_t; h_t]$$

New Reward

Dynamic Completion

Dynamically augment the action space of each entity during reasoning process

Dynamic Anticipation and Completion for Multi-Hop Reasoning over Sparse Knowledge Graph. EMNLP. 2020

DacKGR

Model	FB15K-237-10%			FB15K-237-20%			FB15K-237-50%			NELL23K			WD-singer		
	MRR	@3	@10	MRR	@3	@10	MRR	@3	@10	MRR	@3	@10	MRR	@3	@10
TransE	10.5	15.9	27.9	12.3	18.0	31.3	17.7	23.4	40.4	8.4	10.9	24.7	21.0	32.1	44.6
DisMult	7.4	7.5	16.9	11.3	11.9	24.0	18.0	20.2	38.1	11.6	11.9	23.2	24.4	27.0	39.8
ConvE	24.5	26.2	39.1	26.1	28.3	41.8	31.3	34.2	50.1	27.6	30.1	46.4	44.8	47.8	56.9
TuckER	25.2	27.2	40.4	26.6	28.8	42.8	31.4	34.2	50.1	26.4	28.9	46.7	42.1	47.1	57.1
NeuralLP	7.9	7.2	13.8	11.2	11.2	17.9	18.2	19.2	24.6	12.2	13.1	26.3	31.9	33.4	48.2
NTP	8.3	11.4	16.9	17.3	16.1	21.7	22.2	23.1	30.7	13.2	14.9	24.1	29.2	31.1	44.2
MINERVA	7.8	7.8	12.2	15.9	16.4	22.7	23.0	24.0	31.1	15.0	15.2	25.4	33.5	37.4	44.9
MultiHopKG	13.6	14.6	21.6	23.0	25.2	35.5	29.2	31.7	44.9	17.8	18.8	29.7	35.6	41.1	47.5
CPL	11.1	12.2	16.8	17.5	18.4	25.7	26.4	28.5	36.8	-	-	-	34.2	40.1	46.3
DacKGR (sample)	21.8	**23.9**	**33.7**	**24.7**	**27.2**	**39.1**	**29.3**	**32.0**	45.7	**20.1**	**21.6**	**33.2**	**38.1**	**42.3**	**50.6**
DacKGR (top)	**21.9**	**23.9**	33.5	24.4	27.1	38.9	**29.3**	31.8	**45.8**	19.1	20.0	30.8	37.0	40.5	46.5
DacKGR (avg)	21.5	23.2	33.4	24.2	26.6	38.8	29.1	31.9	45.4	17.1	18.6	28.2	36.4	40.1	48.0

Table 3: Link prediction results on five datasets from Freebase, NELL and Wikidata. @3 and @10 denote Hits@3 and Hits@10 metrics, respectively. All metrics are multiplied by 100. The best score of multi-hop reasoning models is in **bold**, and the best score of embedding-based models is underlined.

Dynamic Anticipation and Completion for Multi-Hop Reasoning over Sparse Knowledge Graph. EMNLP. 2020

TRE (Transitive Relation Embedding)

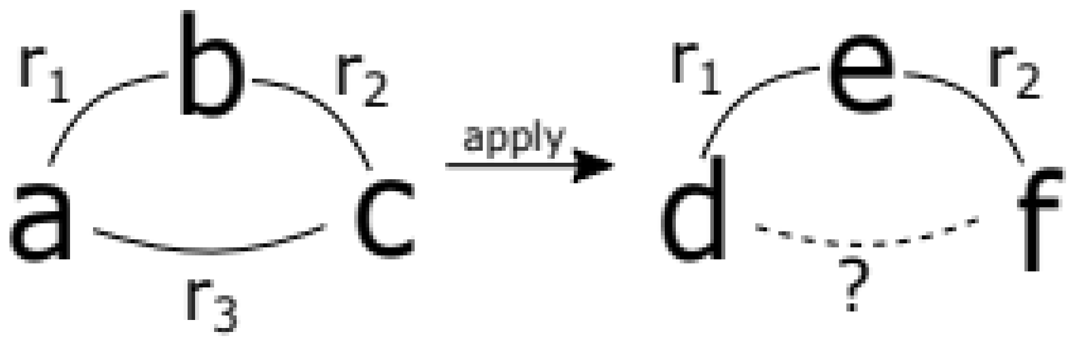
Independent of entities

it learns strictly triangle relation pattern and does not generalize

Only need relation embedding

unable to predict true relations that have never or infrequently appeared in relation patterns

Interpretability



On Completing Sparse Knowledge Base with Transitive Relation Embedding. AAAI. 2019

TRE

$$
\begin{aligned}
Confidence(r_o^+|r_p, r_q) &= \frac{Freqency(r_p, r_q, r_o^+)}{Freqency(r_p, r_q)} \\
Confidence(r_p|r_o^+, r_q) &= \frac{Freqency(r_p, r_q, r_o^+)}{Freqency(r_o^+, r_q)} \\
Confidence(r_q|r_o^+, r_p) &= \frac{Freqency(r_p, r_q, r_o^+)}{Freqency(r_o^+, r_p)} \\
Confidence(r_o^-|r_p, r_q) &= \frac{Freqency(r_p, r_q, r_o^-)}{Freqency(r_p, r_q)} \\
Confidence(r_p|r_o^-, r_q) &= \frac{Freqency(r_p, r_q, r_o^-)}{Freqency(r_o^-, r_p)} \\
Confidence(r_q|r_o^-, r_p) &= \frac{Freqency(r_p, r_q, r_o^-)}{Freqency(r_o^-, r_q)}
\end{aligned} \tag{2}
$$

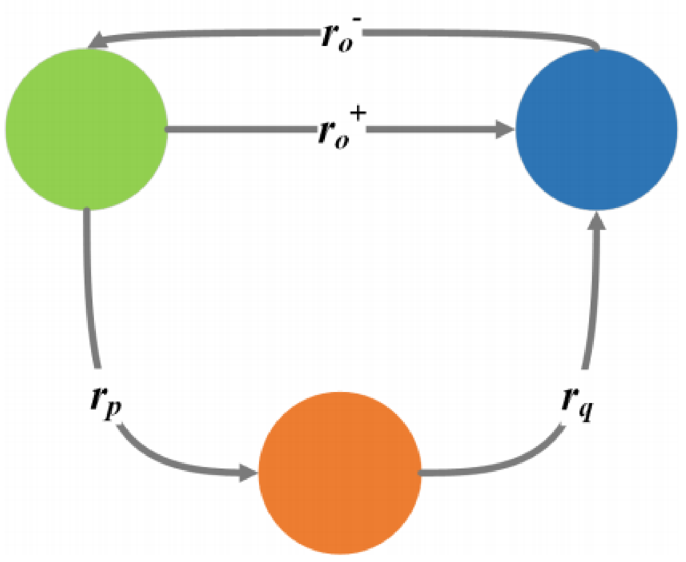


Figure 2: Triangle Pattern

On Completing Sparse Knowledge Base with Transitive Relation Embedding. AAAI. 2019

TRE

$$\begin{aligned}\overrightarrow{V_{r_p,r_q}} &= M_1\overrightarrow{r_p} + M_2\overrightarrow{r_q},\\ \overrightarrow{U^+_{r_o}} &= M_3^+\overrightarrow{r_o},\\ \overrightarrow{U^-_{r_o}} &= M_3^-\overrightarrow{r_o}\end{aligned} \tag{3}$$

$$\begin{aligned}
P(r_o^+|r_p,r_q) &= \frac{exp(\overrightarrow{U^+_{r_o}}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R[exp(\overrightarrow{U^+_{r_k}}^T\overrightarrow{V_{r_p,r_q}}) + exp(\overrightarrow{U^-_{r_k}}^T\overrightarrow{V_{r_p,r_q}})]}\\
P(r_o^-|r_p,r_q) &= \frac{exp(\overrightarrow{U^-_{r_o}}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R[exp(\overrightarrow{U^+_{r_k}}^T\overrightarrow{V_{r_p,r_q}}) + exp(\overrightarrow{U^-_{r_k}}^T\overrightarrow{V_{r_p,r_q}})]}\\
P(r_p|r_o^+,r_q) &= \frac{exp(\overrightarrow{U^+_{r_o}}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R exp(\overrightarrow{U^+_{r_o}}^T\overrightarrow{V_{r_k,r_q}})}\\
P(r_p|r_o^-,r_q) &= \frac{exp(\overrightarrow{U^-_{r_o}}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R exp(\overrightarrow{U^-_{r_o}}^T\overrightarrow{V_{r_k,r_q}})}\\
P(r_q|r_o^+,r_p) &= \frac{exp(\overrightarrow{U^+_{r_o}}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R exp(\overrightarrow{U^+_{r_o}}^T\overrightarrow{V_{r_p,r_k}})}\\
P(r_q|r_o^-,r_p) &= \frac{exp(\overrightarrow{U^-_{r_o}}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R exp(\overrightarrow{U^-_{r_o}}^T\overrightarrow{V_{r_p,r_k}})}
\end{aligned} \tag{4}$$

On Completing Sparse Knowledge Base with Transitive Relation Embedding. AAAI. 2019

IterE

Embedding Learning

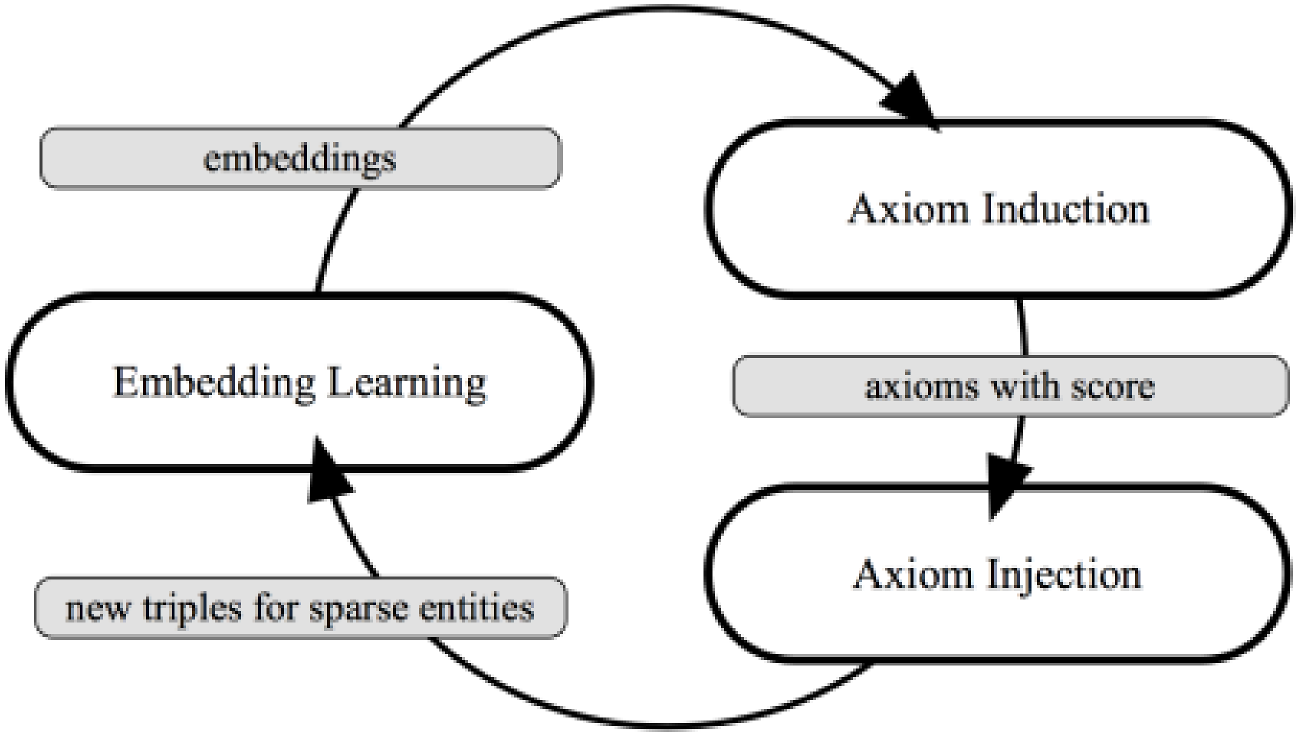
Learn a awesome embedding model

Axiom Induction

Induce a set of axioms

Axiom Injection

New facts



Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning. WWW. 2019

IterE

Traverse and Selection

Step 1: Generate Axioms

Object Property Axioms	Rule Form	According to Linear Map Assumption	Rule Conclusion
$\mathtt{ReflexiveOP}(r)$	(x, r, x)	$\mathbf{v}_x\mathbf{M}_r = \mathbf{v}_x$	$\mathbf{M}_r = \mathbf{I}$
$\mathtt{SymmetricOP}(r)$	$(y, r, x) \leftarrow (x, r, y)$	$\mathbf{v}_y\mathbf{M}_r = \mathbf{v}_x; \mathbf{v}_x\mathbf{M}_r = \mathbf{v}_y$	$\mathbf{M}_r\mathbf{M}_r = \mathbf{I}$
$\mathtt{TransitiveOP}(r)$	$(x, r, z) \leftarrow (x, r, y), (y, r, z)$	$\mathbf{v}_x\mathbf{M}_r = \mathbf{v}_z; \mathbf{v}_x\mathbf{M}_r = \mathbf{v}_y, \ \mathbf{v}_y\mathbf{M}_r = \mathbf{v}_z,$	$\mathbf{M}_r\mathbf{M}_r = \mathbf{M}_r$
$\mathtt{EquivalentOP}(r_1, r_2)$	$(x, r_2, y) \leftarrow (x, r_1, y)$	$\mathbf{v}_x\mathbf{M}_{r_2} = \mathbf{v}_y, \ \mathbf{v}_x\mathbf{M}_{r_1} = \mathbf{v}_y$	$\mathbf{M}_{r_1} = \mathbf{M}_{r_2}$
$\mathtt{subOP}(r_1, r_2)$	$(x, r_2, y) \leftarrow (x, r_1, y)$	$\mathbf{v}_x\mathbf{M}_{r_2} = \mathbf{v}_y, \ \mathbf{v}_x\mathbf{M}_{r_1} = \mathbf{v}_y$	$\mathbf{M}_{r_1} = \mathbf{M}_{r_2}$
$\mathtt{inverseOP}(r_1, r_2)$	$(x, r_1, y) \leftarrow (y, r_2, x)$	$\mathbf{v}_x\mathbf{M}_{r_1} = \mathbf{v}_y, \ \mathbf{v}_y\mathbf{M}_{r_2} = \mathbf{v}_x$	$\mathbf{M}_{r_1}\mathbf{M}_{r_2} = \mathbf{I}$
$\mathtt{subOP}(\mathtt{OPChain}(r_1, r_2), r)$	$(y_0, r, y_2) \leftarrow (y_0, r_1, y_1), (y_1, r_2, y_2)$	$\mathbf{v}_{y_0}\mathbf{M}_r = \mathbf{v}_{y_2}, \ \mathbf{v}_{y_0}\mathbf{M}_{r_1} = \mathbf{v}_{y_1}, \ \mathbf{v}_{y_1}\mathbf{M}_{r_2} = \mathbf{v}_{y_2}$	$\mathbf{M}_{r_1}\mathbf{M}_{r_2} = \mathbf{M}_r$

$$\mathrm{A}(r, Var\{r', r''\})$$

Step 2: Complete Axioms

select k facts (e', r, e'') related with r, replace r', r'' with the relations that directly link to e', e''

Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning. WWW. 2019

IterE

Table 5: Link prediction results with MRR and Hit@n on WN18RR-sparse and FB15k-237-sparse. Underlined scores are the better ones between ANALOGY and IterE(ANALOGY). Boldface scores are the best results among all methods.

	WN18-sparse					FB15k-sparse				
	MRR (filter)	MRR (raw)	Hit@1 (filter)	Hit@3 (filter)	Hit10 (filter)	MRR (filter)	MRR (raw)	Hit@1 (filter)	Hit@3 (filter)	Hit10 (filter)
TransE[3]	41.8	33.5	10.2	71.1	84.7	39.8	25.5	25.8	48.6	64.5
DistMult[47]	73.8	55.8	59.3	87.5	93.1	60.0	32.4	**61.8**	65.1	75.9
ComplEx[35]	91.1	67.7	89.0	93.3	94.4	61.6	32.7	54.0	65.7	76.1
ANALOGY[23]	**91.3**	67.5	89.0	93.4	94.4	62.0	33.1	54.3	66.1	76.3
IterE(ANALOGY)	90.1	67.5	87.0	93.1	**94.8**	61.3	35.9	52.9	66.2	76.7
IterE(ANALOGY) + **axioms**	**91.3**	**78.9**	**89.1**	**93.5**	**94.8**	**62.8**	**38.8**	55.1	**67.3**	**77.1**

	WN18RR-sparse					FB15k-237-sparse				
	MRR (filter)	MRR (raw)	Hit@1 (filter)	Hit@3 (filter)	Hit10 (filter)	MRR (filter)	MRR (raw)	Hit@1 (filter)	Hit@3 (filter)	Hit10 (filter)
TransE[3]	14.6	12.4	3.4	24.7	28.8	23.8	15.6	16.4	26.1	38.5
DistMult[47]	25.5	20.8	23.8	26.0	22.5	20.4	12.9	12.8	22.6	36.2
ComplEx[35]	25.9	21.4	24.6	26.2	28.6	19.7	13.3	12.0	21.7	35.4
ANALOGY[23]	19.8	13.3	24.6	27.5	28.7	19.8	13.9	12.3	21.4	34.9
IterE(ANALOGY)	27.2	22.7	25.0	**28.1**	**31.4**	20.7	14.0	13.1	22.8	36.2
IterE(ANALOGY) + **axioms**	**27.4**	**25.7**	**25.4**	**28.1**	**31.4**	**24.7**	**18.6**	**17.9**	**26.2**	**39.2**

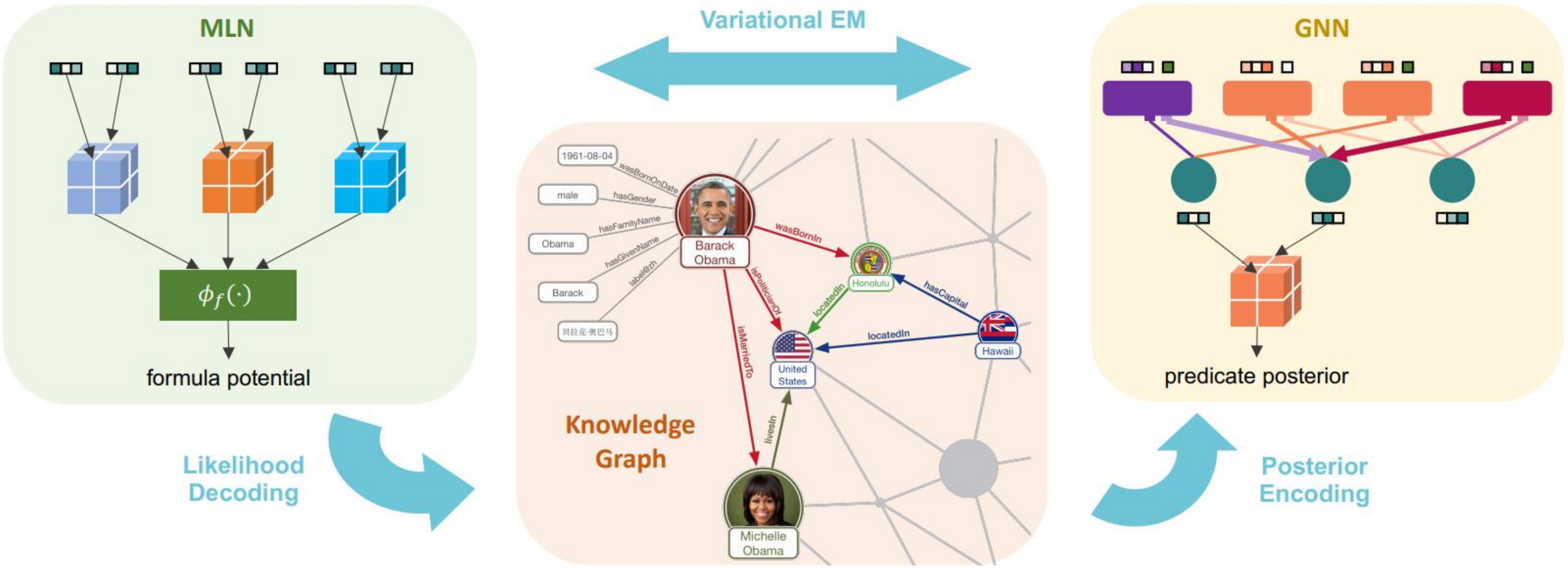
pLogic

$$p_w(O,H) = \frac{1}{Z} exp(\sum_l w_l n_l(O,H))$$

$$log p_w(O) \geq E_{q_\theta(H)}[log p_w(O,H) - log q_\theta(H)]$$

Probabilistic Logic Neural Networks for Reasoning. NIPS. 2019

ExpressGNN*



Efficient Probabilistic Logic Reasoning with Graph Neural Networks. ICLR. 2020

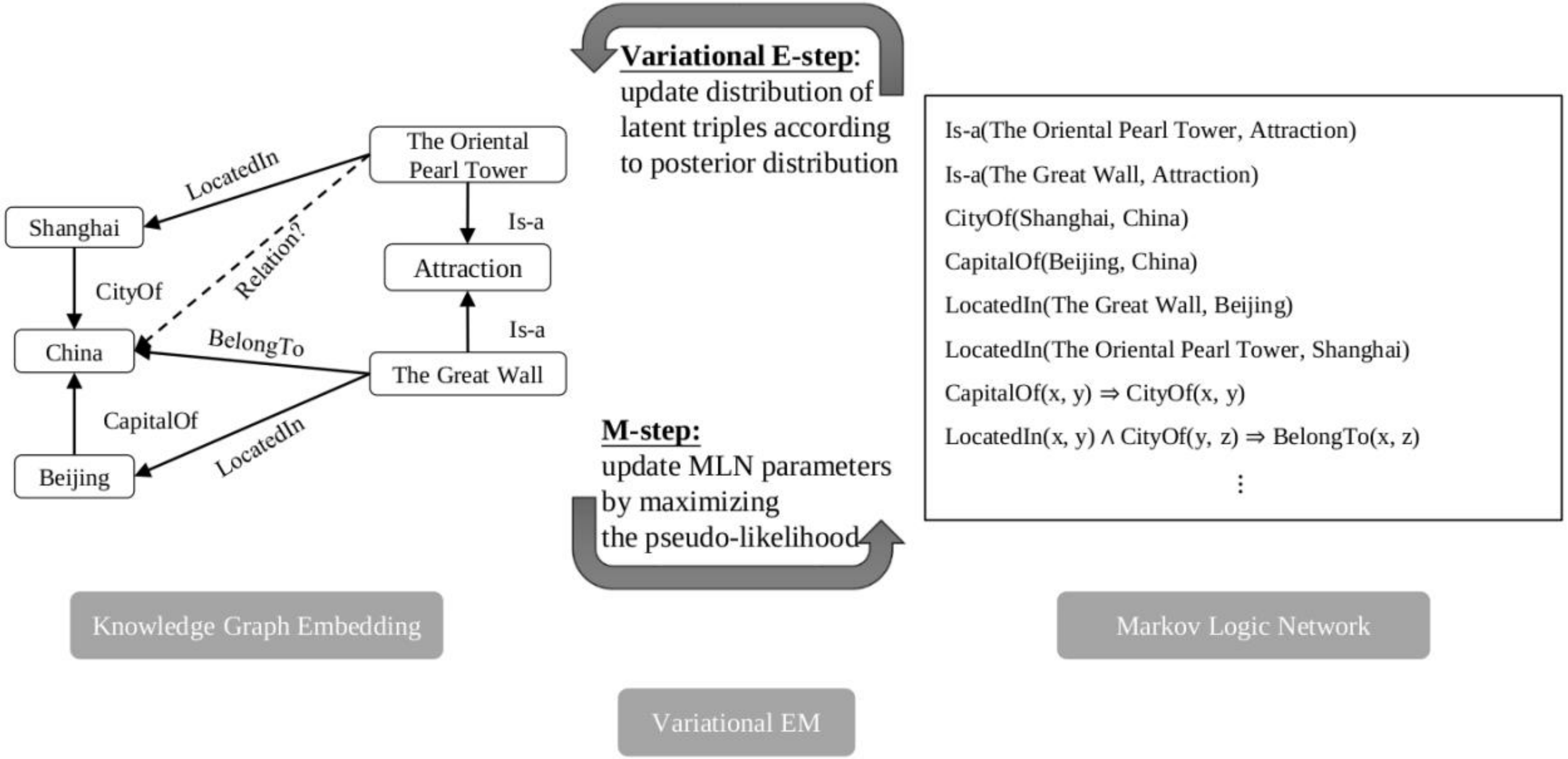
ExpressGNN*

Table 3: Performance on FB15K-237 with varied training set size.

Model	MRR					Hits@10				
	0%	5%	10%	20%	100%	0%	5%	10%	20%	100%
MLN	-	-	-	-	0.10	-	-	-	-	16.0
NTN	0.09	0.10	0.10	0.11	0.13	17.9	19.3	19.1	19.6	23.9
Neural LP	0.01	0.13	0.15	0.16	0.24	1.5	23.2	24.7	26.4	36.2
DistMult	0.23	0.24	0.24	0.24	0.31	40.0	40.4	40.7	41.4	48.5
ComplEx	0.24	0.24	0.24	0.25	0.32	41.1	41.3	41.9	42.5	51.1
TransE	0.24	0.25	0.25	0.25	0.33	42.7	43.1	43.4	43.9	52.7
RotatE	0.25	0.25	0.25	0.26	0.34	42.6	43.0	43.5	44.1	53.1
pLogicNet	-	-	-	-	0.33	-	-	-	-	52.8
ExpressGNN-E	**0.42**	**0.42**	0.42	0.44	0.45	53.1	53.1	53.3	55.2	57.3
ExpressGNN-EM	**0.42**	**0.42**	**0.43**	**0.45**	**0.49**	**53.8**	**54.6**	**55.3**	**55.6**	**60.8**

Efficient Probabilistic Logic Reasoning with Graph Neural Networks. ICLR. 2020

pGAT



Probabilistic Logic Graph Attention Networks for Reasoning. WWW. 2020

pGAT

Table 1: Results of link prediction on test sets of FB15K-237 and WN18RR respectively. The best scores are in bold. The second best scores are underlined.

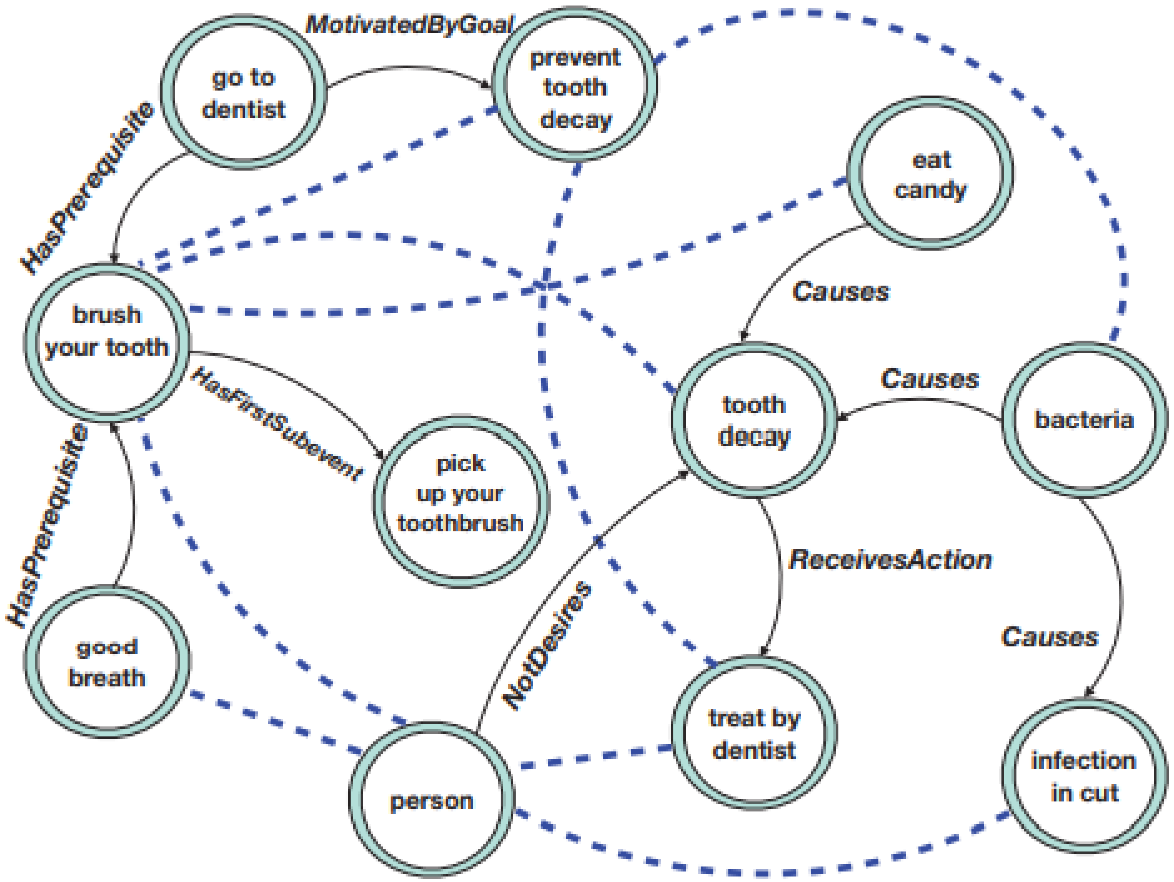
Method	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE [5]	323	0.279	19.8	37.6	44.1	2300	0.243	4.27	44.1	53.2
DistMult [27]	512	0.281	19.9	30.1	44.6	7000	0.444	41.2	47	50.4
ComplEx [24]	546	0.278	19.4	29.7	45	7882	0.449	40.9	46.9	50.4
RotatE [20]	185	0.297	20.5	32.8	48.0	3277	**0.470**	**42.2**	<u>48.8</u>	56.5
ConvE [8]	245	0.312	22.5	34.1	49.7	4464	0.456	<u>41.9</u>	47	53.1
ConvKB [13]	216	0.289	19.8	32.4	47.1	**1295**	0.265	5.82	44.5	55.8
R-GCN[19]	600	0.164	10	18.1	30	6700	0.123	20.7	13.7	8
KBAT [11]	204	<u>0.431</u>	<u>35.5</u>	<u>46.2</u>	<u>57.8</u>	1970	0.431	35.2	47.3	<u>57.4</u>
BLP [7]	1985	0.092	6.2	9.8	15.0	12051	0.254	18.7	31.3	35.8
MLN [17]	1980	0.098	6.7	10.3	16.0	11549	0.259	19.1	32.2	36.1
pLogicNet [16]	173	0.330	23.1	36.9	52.8	3436	0.230	1.5	41.1	53.1
pLogicNet* [16]	**173**	0.332	23.7	36.7	52.4	3408	0.441	39.8	44.6	53.7
pGAT	<u>181</u>	**0.457**	**37.7**	**49.4**	**60.9**	<u>1868</u>	<u>0.459</u>	39.5	**48.9**	**57.8**

Probabilistic Logic Graph Attention Networks for Reasoning. WWW. 2020

MCC

Graph Densification and GCN:
add synthetic edge *sim*
learning from graph structure

Knowledge from PTM
transfer learning from language to knowledge graphs



Commonsense Knowledge Base Completion with Structural and Semantic Context. AAAI. 2020

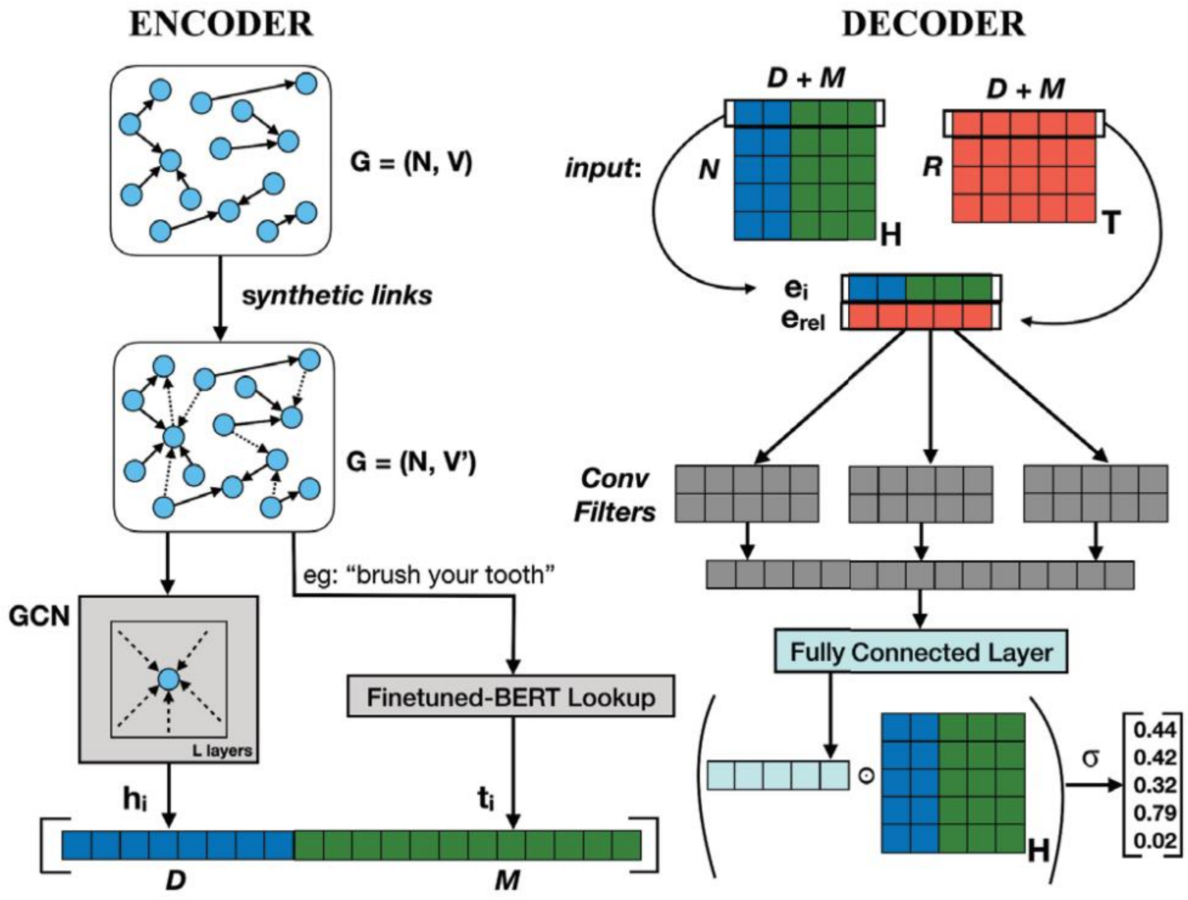
MCC

Knowledge from PTM
Phrase from natural language

Graph structure
Relation-aware GCN + Att

Densification:
Add synthetic edge <u>*sim*</u> depend on PTM

Fusion:
Progressive masking



Commonsense Knowledge Base Completion with Structural and Semantic Context. AAAI. 2020

MCC

	CN-100K				ATOMIC			
	MRR	HITS@1	@3	@10	MRR	HITS@1	@3	@10
DISTMULT	8.97	4.51	9.76	17.44	12.39	9.24	15.18	18.30
COMPLEX	11.40	7.42	12.45	19.01	**14.24**	**13.27**	14.13	15.96
CONVE	20.88	13.97	22.91	34.02	10.07	8.24	10.29	13.37
CONVTRANSE	18.68	7.87	23.87	38.95	12.94	12.92	12.95	12.98
COMET-NORMALIZED	6.07	0.08	2.92	21.17	3.36*	0.00*	2.15*	15.75*
COMET-TOTAL	6.21	0.00	0.00	24.00	4.91*	0.00*	2.40*	21.60*
BERT + CONVTRANSE	49.56	38.12	55.5	71.54	12.33	10.21	12.78	16.20
GCN + CONVTRANSE	29.80	21.25	33.04	47.50	13.12	10.70	13.74	17.68
SIM + GCN + CONVTRANSE	30.03	21.33	33.46	46.75	13.88	11.50	**14.44**	**18.38**
GCN + BERT + CONVTRANSE	50.38	38.79	56.46	72.96	10.8	9.04	11.21	14.10
SIM + GCN + BERT + CONVTRANSE	**51.11**	**39.42**	**59.58**	**73.59**	10.33	8.41	10.79	13.86

Commonsense Knowledge Base Completion with Structural and Semantic Context. AAAI. 2020

Future Work

Entity-independent Settings and Inductive Settings

Explore efficient ways to find rules or grounding