RLogic: Recursive Logical Rule Learning from Knowledge Graphs

2024-09-13

presenter : Sooho Moon

DMAIS

Kewei Cheng, Wei Wang, Yizhou Sun
Department of Computer Science, University of California, Los Angeles, Los Angeles, CA

Jiahao Liu
Department of Computer Science, Brown University, Providence, RI

KDD 2022

INDEX

Main interest

Problem with previous models

Architecture

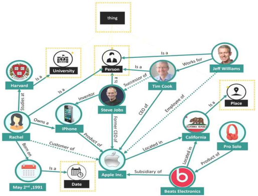
Experiments

Conclusion

Main interest

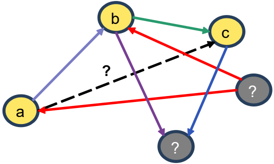
- **Knowledge Graph(KG)**

  실세계 데이터를 triple(i.e. (h, r, t) 또는 r(h, t))
  단위로 저장하는 그래프 구조



- **Knowledge Graph Reasoning**

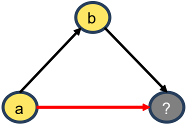
  KG를 입력 받아 여러 추론 task를
  수행할 수 있는 모델을 만드는 과제



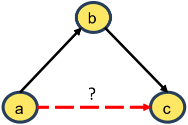
Main interest

- **Logical rule mining**

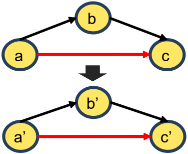
KG 전반에 깔려있는 **일반적인 지식**을 **규칙(rule)**으로 추출하고자 함



KG completion



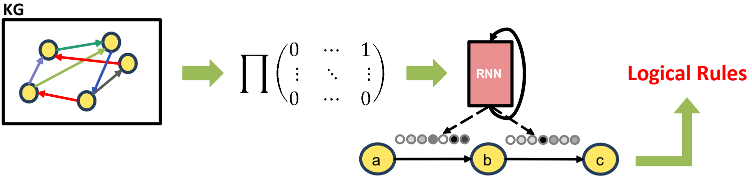
Head prediction



Inductive link prediction

Problem with previous models

기존의 접근들은 단순히 KG에서 관찰되는 데이터들만을 이용해 rule을 평가, 학습
(e.g., NeuralLP(NeurIPS, 17), DRUM(NeurIPS, 19), RNNLogic(ICLR, 21))

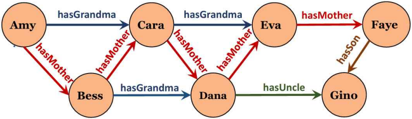


* 이 접근은 구체적으로 존재하는 경로에 대해서만 학습 가능함

Problem with previous models

하지만 KG는 sparse 하기 때문에 relation link가 모두 존재하기를 바랄 수 없음

Knowledge Graph



rule head(r_h) = **hasGrandma**

body instance(r_b) : **hasMother ^ hasMother**

rule instance : **head ← body instance**

기존의 method들은 rule을 학습하기 위해서 rule instance의 support가 필요하므로
hasUncle* ← *hasGrandma* ∧ *hasSon 을 학습하지 못함

Problem with previous models

- Rule을 평가하는 score을 구성할 때 완전히 rule instance에 의존함

  → KG의 sparsity에 대응할 수 없음

- Logical rule이 독립적이라고 가정하고 rule을 학습함

  → RLogic에서 중요시 여기는 deductive nature와 상반되는 과거 모델들이 따르는 가정임

  * **deductive nature** : 새로운 rule 학습에 있어 학습된 rule을 사용함

Problem with previous models

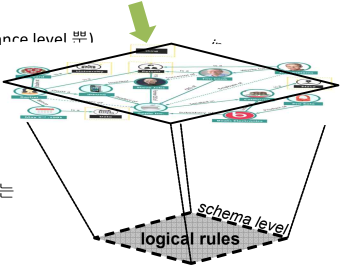
- **Problem definition**

(Logical rule은 schema level 개념이지만 증거는 instance level 뿐)

* Schema level에서 직접 학습하기 위해
representation learning-based model을 구축

* 각 rule $(r_h, \boldsymbol{r_b})$에 대해 타당한 점수를 매길 수 있는
score function을 모델링

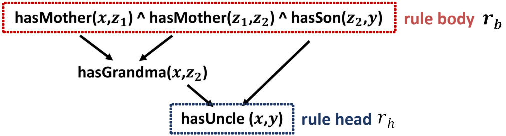
obvervation



Architecture

- **Rule body와 rule head의 관계를 나타내는 방법**

Deductive nature에 의해 긴 sequence의 path도 재귀적으로 짧은 path로 치환 가능



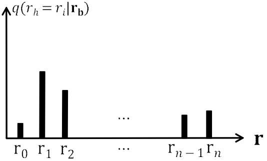
즉 2개의 relation을 하나의 relation으로 대체할 수 있는 모델만 있으면
어떤 길이의 **body**도 하나의 **head**로 표현 가능

Architecture

- **Relation Path Encoder(q)**

$$s(r_h, \mathbf{r_b}) = q(r_h = r_i | \mathbf{r_b})$$

: **body**가 **head**로 표현될 확률



e.g., $q(r_h | r_{b_1}, r_{b_2}, r_{b_3}) = \sum_k q(r_h | r_k, r_{b_3}) q(r_k | r_{b_1}, r_{b_2})$ (body length 3을 하나의 head로)

재귀적으로 body를 head로 대체하므로 RNN같은 무거운 sequence 모델을 대체함

Architecture

▪ Close Ratio Predictor(p)

Deduct된 head가 KG의 sparsity 때문에 언제나 관찰 가능하다는 보장이 없음

➡ unknown region에서의 prediction을 penalize할 위험이 큼

이 문제를 줄여보고자 path($\boldsymbol{r_b}$)가 닫힐 확률을 계산함(r_t = target relation = 실제 head)

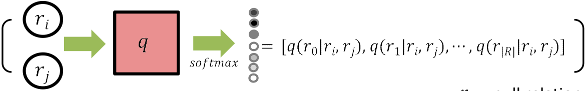
$$p(r_t|\mathbf{r_b}) = \sum_h p(r_t|r_h)q(r_h|\mathbf{r_b})$$

이로써 logical rule에 의한 “ideal prediction”과 KG의 “real observation” 사이의 간극을 줄임

Architecture

Relation Path Encoder : 2개의 relation을 어떤 relation으로 줄일지에 대한 확률

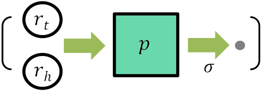
$q(r_h|r_i, r_j)$



r_0 : null relation

Closed Ratio Predictor : 최종적으로 body가 target으로 닫힐 확률

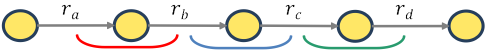
$p(r_t|r_h)$



Architecture

- **Relation Path Encoder(q) - deduction 순서**

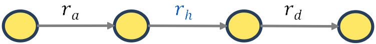
$\boldsymbol{r_b}$:



$$E((r_i, r_j)) = \sum_{r_h \in R} -q(r_h | r_i, r_j) \log q(r_h | r_i, r_j)$$

e.g., $E((r_b, r_c)) < E((r_a, r_b)), E((r_c, r_d))$

$\boldsymbol{r_b}$:

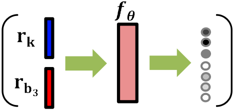


Architecture

- **Relation Path Encoder - relation pair to single head**

$\mathbf{r}_\mathrm{i}$: representation embedding vector, training 과정에서 학습됨

$$q(r_h|r_{b_1}, r_{b_2}, r_{b_3}) = \sum_k q(r_k|r_{b_1}, r_{b_2}) f_\theta(\mathbf{r}_\mathrm{k}, \mathbf{r}_{\mathrm{b}_3})$$



approximate

$$f_\theta(\tilde{\mathbf{r}}, \mathbf{r}_{\mathrm{b}_3}) \quad \left[\tilde{\mathbf{r}} = \sum_k q(r_k|r_{b_1}, r_{b_2}) \cdot \mathbf{r}_\mathrm{k}\right]$$

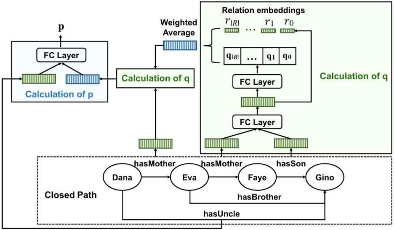
weighted average

Architecture

- **Close Ratio Predictor**

Relation path encoder에서 얻은 $\tilde{\mathbf{r}}$와 $\mathbf{r}_t$ 를 FC Layer에 입력

- **Total RLogic framework**



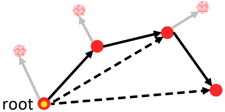
*** top 2400 rules are selected according ONLY from q**

*** p just indirectly helps q**

Architecture

- **Model Training**

Closed path sampler for training data(random walk)



Objective function for **relation path encoder**

$$\sum_{(r_h,\mathbf{r}_\mathbf{b})\in\mathcal{P}} \sum_{(r_h{}',\mathbf{r}_\mathbf{b})\in\mathcal{N}(r_h,\mathbf{r}_\mathbf{b})} [q(r_h|\mathbf{r}_\mathbf{b}) + \gamma - q(r_h{}'|\mathbf{r}_\mathbf{b}))]_+$$

* KG operate under
Open World Assumption(OWA)

Objective function for **closed ratio predictor**

$$\sum_{(r_t,r_h)\in\mathcal{P}} \log p(r_t|r_h) + \sum_{(r_t,r_h)\in\mathcal{N}} \log(1 - (p(r_t|r_h))$$

Experiments

- **Datasets and metrics for evaluation**

	KG Completion	Rule Quality Checking
WN18RR	O	
FB15k-237	O	O
YAGO3-10	O	
Family		O

Dataset	# Data	# Relation	# Entity
FB15K-237	310,116	237	14,541
WN18RR	93,003	11	40,943
YAGO3-10	1,089,040	37	123,182
Family	28,356	12	3007

filtered setting

+

MRR(Mean Reciprocal Rank), Hits@1, Hits@10, MAP(Mean Average Percision)(all higher, the better)

+

random protocal for same score ranking

Experiments

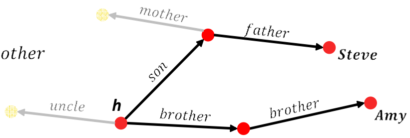
- **How logical rule is used for prediction**

Forward chaining via sparse matrix multiplication to nevigate to candidate answers

$$\sum_{\delta \in \text{rule space}} \sum_{\text{path} \in P(h,r,e)} s(\delta)$$

head : ***brother***
rules : $son \wedge father, brother \wedge brother$

(h, brother, ?)



Experiments

- **KG Completion(with transductive based models)**

Category	Model	WN18RR			FB15K-237			YAGO3-10		
		MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
KGE	TransE	0.23	2.2	52.4	0.29	18.9	46.5	0.36	25.1	58.0
	DistMult	0.42	38.2	50.7	0.22	13.6	38.8	0.34	24.3	53.3
	ConvE	0.43	40.1	52.5	**0.32**	21.6	50.1	0.36	26.5	55.6
	ComplEx	0.44	41.0	51.2	0.24	15.8	42.8	0.34	24.8	54.9
	RotatE	**0.47**	42.9	**55.7**	**0.32**	**22.8**	**52.1**	**0.49**	**40.2**	**67.0**
Rule Learning	Neural-LP†	0.38	36.8	40.8	0.24	17.3	36.2	-	-	-
	NLIL†	0.30	20.1	33.5	0.25	13.8	32.4	-	-	-
	DRUM†	0.38	36.9	41.0	0.23	17.4	36.4	-	-	-
	AMIE	0.36	39.1	48.5	0.23	14.8	41.9	0.25	20.6	34.3
	RNNLogic (w/o emb)‡	0.46	41.4	53.1	0.29	20.8	44.5	-	-	-
	RLogic	**0.47**	**44.3**	53.7	0.31	20.3	50.1	0.36	25.2	50.4

† Neural-LP, NLIL and DRUM exceeds the capacity of our machines on YAGO3-10 dataset

‡ Results on RLogic are taken from the original papers.

Experiments

- **Inductive link prediction(KG Completion)**

Model	WN18RR			FB15K-237			YAGO3-10		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
KGE†	-	-	-	-	-	-	-	-	-
Neural-LP‡	0.23	20.3	33.1	0.14	9.3	27.6	-	-	-
DRUM‡	0.23	20.5	34.4	0.16	10.8	29.3	-	-	-
AMIE	0.32	33.6	45.5	0.19	13.9	38.0	0.21	15.8	30.1
RLogic	**0.43**	**42.1**	**50.8**	**0.29**	**18.4**	**48.7**	**0.32**	**22.8**	**47.2**

† KGE methods are not applicable in inductive setting.
‡ Neural-LP, NLIL and DRUM exceeds the capacity of our machines on YAGO3-10 dataset
* Results on RLogic are taken from the original papers.

training, validation, test = 7:1:2 with disjoint entities

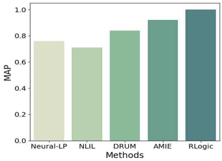
Model	WN18RR			FB15K-237			YAGO3-10		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
RNNLogic+ (with emb.)†	0.51	**47.1**	59.7	0.35	25.8	53.3	-	-	-
RLogic+	**0.52**	46.6	**60.4**	**0.55**	**51.1**	**64.3**	**0.53**	**42.6**	**70.3**
RotatE	**0.47**	<u>42.9</u>	**55.7**	**0.32**	**22.8**	**52.1**	**0.49**	**40.2**	**67.0**

RLogic+ : variant of RLogic specifically for KG completion tasks

Experiments

- **Rule quality check via rule head prediction tasks**

Neural-LP	RLogic
$\text{brother}(x,y) \leftarrow \text{inv\_sister}(x,y)$	$\text{brother}(x,y) \leftarrow \text{brother}(x,z) \land \text{sister}(z,y)$
$\text{brother}(x,y) \leftarrow \text{sister}(x,z) \land \text{inv\_brother}(z,y)$	$\text{brother}(x,y) \leftarrow \text{son}(x,z) \land \text{father}(z,y)$
$\text{brother}(x,y) \leftarrow \text{inv\_sister}(x,z) \land \text{inv\_sister}(z,y)$	$\text{brother}(x,y) \leftarrow \text{brother}(x,z) \land \text{brother}(z,y)$
$\text{wife}(x,y) \leftarrow \text{inv\_husband}(x,z) \land \text{inv\_husband}(z,y)$	$\text{wife}(x,y) \leftarrow \text{mother}(x,z) \land \text{inv\_father}(z,y)$
$\text{wife}(x,y) \leftarrow \text{inv\_husband}(x,y)$	$\text{wife}(x,y) \leftarrow \text{mother}(x,z) \land \text{daughter}(z,y)$
$\text{wife}(x,y) \leftarrow \text{inv\_husband}(x,z) \land \text{daughter}(z,y)$	$\text{wife}(x,y) \leftarrow \text{mother}(x,z) \land \text{son}(z,y)$
$\text{son}(x,y) \leftarrow \text{brother}(x,y) \land \text{inv\_mother}(z,y)$	$\text{son}(x,y) \leftarrow \text{brother}(x,z) \land \text{inv\_father}(z,y)$
$\text{son}(x,y) \leftarrow \text{inv\_mother}(x,z) \land \text{inv\_mother}(z,y)$	$\text{son}(x,y) \leftarrow \text{brother}(x,z) \land \text{son}(z,y)$
$\text{son}(x,y) \leftarrow \text{brother}(x,y)$	$\text{son}(x,y) \leftarrow \text{son}(x,z) \land \text{inv\_wife}(z,y)$



top 3 rules for relation *brother, wife, son*

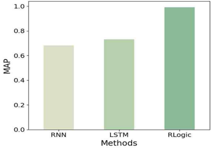
blue : not always true rule

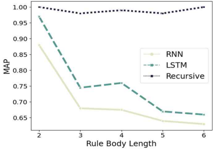
red : incorrect rule

(Human annotation are used for ground truth evaluation)

Experiments

- **Relation path encode → RNN, LSTM**





Query 1 $?(x,y) \leftarrow hasBrother(x,z_1) \wedge hasSister(z_1,y)$

Query 2 $?(x,y) \leftarrow hasBrother(x,z_1) \wedge hasSister(z_1,z_2) \wedge hasFather(z_2,z_3) \wedge hasWife(z_3,z_4) \wedge hasHusband(z_4,y)$

Query 3 $?(x,y) \leftarrow hasBrother(x,z_1) \wedge hasSister(z_1,z_2) \wedge hasFather(z_2,z_3) \wedge hasWife(z_3,z_4) \wedge hasHusband(z_4,z_5) \wedge hasBrother(z_5,y)$

Query	RNN	LSTM	RLogic
Query 1	**hasSister 0.7** hasBrother 0.1 hasMother 0.1	**hasSister 0.9** hasBrother 0.0 hasNiece 0.0	**hasSister 0.9** hasDaughter 0.0 hasBrother 0.0
Query 2	hasSister 0.8 **hasFather 0.1** hasUncle 0.0	**hasFather 0.8** hasMother 0.1 hasDaughter 0.1	**hasFather 1.0** hasSon 0.0 hasMother 0.0
Query 3	hasBrother 0.3 hasDaughter 0.3 hasSon 0.2	hasMother 0.8 hasBrother 0.1 hasFather 0.1	**hasUncle 0.8** hasBrother 0.1 hasAunt 0.0

Deductive nature를 따르는 RLogic의 relation path encoder가 RNN, LSTM보다 rule 구성에 있어 압도적인 성능을 보임

Experiments

▪ Conclusion

- KG의 rule instance에만 강하게 의존하지 않는 representation learning based scoring model을 정의함으로써 일반화가 부족했던 기존의 모델들의 한계를 극복함

- Deductive nature로 긴 sequence의 body를 재귀적으로 작은 부분으로 쪼개어 supporting evidence가 미약해도 logical rule을 학습할 수 있게 해줌

Thank You!

Contact: Sooho Moon (Email: moonwalk725@cau.ac.kr)