Random Walk Inference and Learning in A Large Scale Knowledge Base

Published as a conference paper at EMNLP 2011

Ni Lao, Tom Mitchell, William W. Cohen

2024-11-01

HoonUi Lee

Index

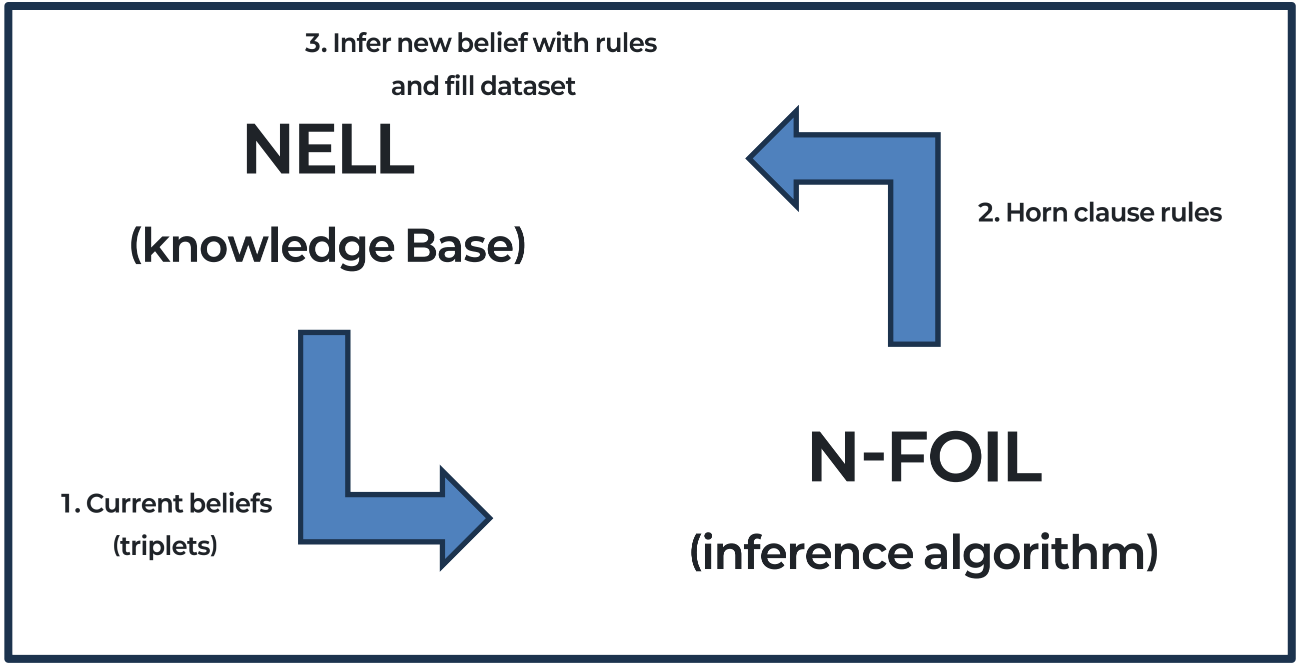
Previous work

PRA

Experiment

Conclusion

Previous work



Previous work

- N-FOIL

❖ **Learn horn clause rule and fill the knowledge base's imperfect part**

- NELL configures knowledge base with data from the web

➢ There's some imperfect part in data from the web

➢ Approaches about how to fill this part

Previous work

- N-FOIL

❖ N-FOIL Algorithm

- Takes as input a set of **positive and negative examples**
- **"Separate-And-Conquer"** strategy to learn Horn clauses

➢ When rule covers many positive examples as possible
and few negative examples as possible, then rule is finally learned

➢ Remove the sample covered by that rule from the training set
➢ To learn new rule about left examples

Input:

+ AtheletePlaysInLeague(HinesWard, NFL)

- AtheletePlaysInLeague(HinesWard, NBA)

General rule :

All Athelete Plays In Sports League

Specialize rule :

AthletePlaysForTeam(a, b) ∧ TeamPlaysInLeague(b, c)

⇒AthletePlaysInLeague(a, c)

Previous work

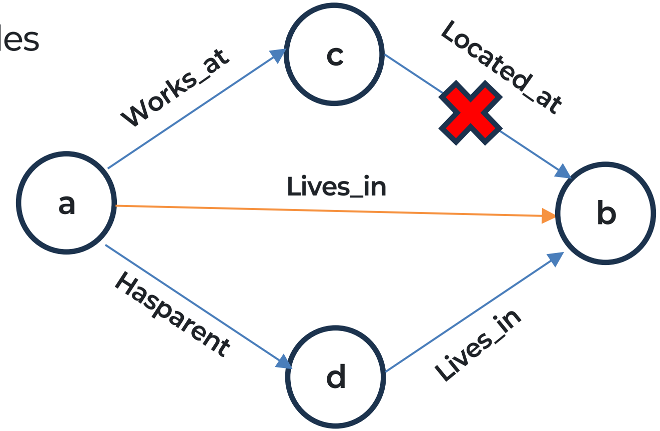
- N-FOIL

❖ Restrictive inference

Learned rule:

Works_at (a, c) ∧ Located_at (c, b)

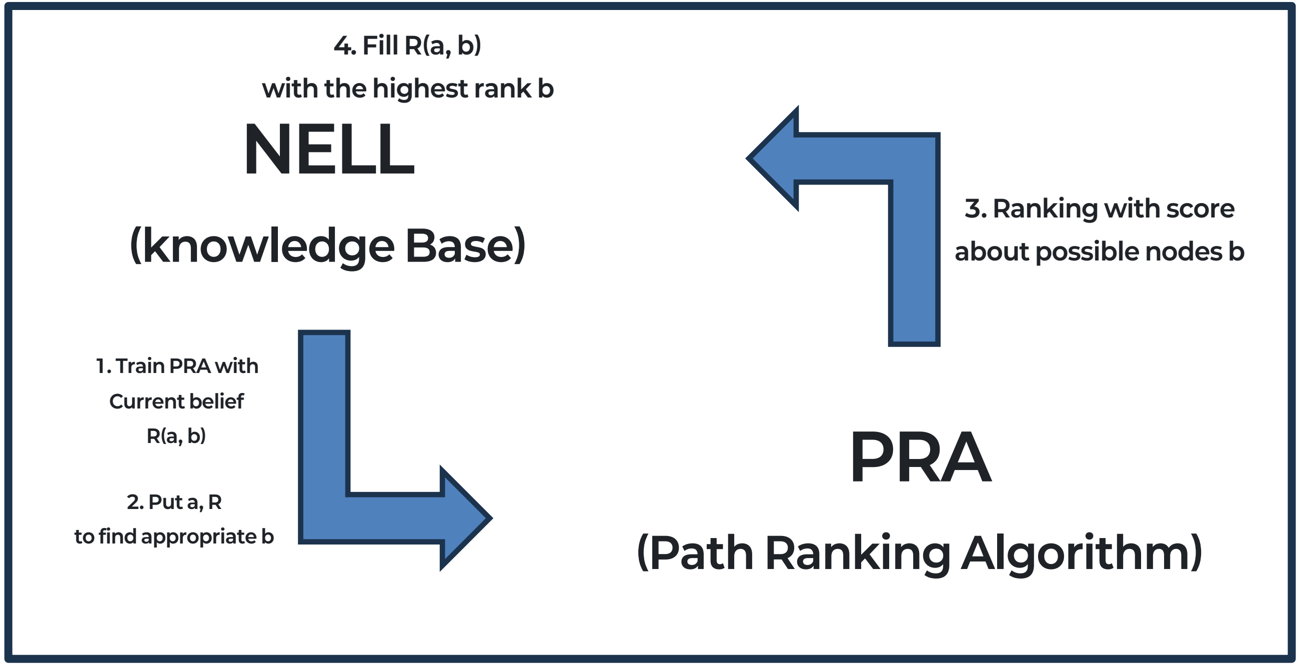
-> Lives_in (a, b)



- N-FOIL consider one rule (path) to infer connection between nodes

- If there exist less learned rule between nodes and the data is imperfect in knowledge base, it is difficult to make reliable inference

PRA


4. Fill R(a, b)
with the highest rank b
NELL
(knowledge Base)
3. Ranking with score
about possible nodes b
1. Train PRA with
Current belief
R(a, b)
2. Put a, R
to find appropriate b
PRA
(Path Ranking Algorithm)

PRA

Which node e
to connect by R ?

$s \xrightarrow{R} e$

Experts
Expert 1 (r_1, r_3)
Expert 2 (r_2, r_4, r_5)
Expert 3 (r_1, r_4, r_5)
Expert 4 (r_2, r_3)
...

e1: $\theta_1 h_{s,P_1}(e) + \theta_2 h_{s,P_2}(e) + \ldots \theta_n h_{s,P_n}(e)$

e2: $\theta_1 h_{s,P_1}(e) + \theta_2 h_{s,P_2}(e) + \ldots \theta_n h_{s,P_n}(e)$

e3: $\theta_1 h_{s,P_1}(e) + \theta_2 h_{s,P_2}(e) + \ldots \theta_n h_{s,P_n}(e)$

e4: $\theta_1 h_{s,P_1}(e) + \theta_2 h_{s,P_2}(e) + \ldots \theta_n h_{s,P_n}(e)$

...

Ranking

1st. e2

2nd. e3

3rd. e1

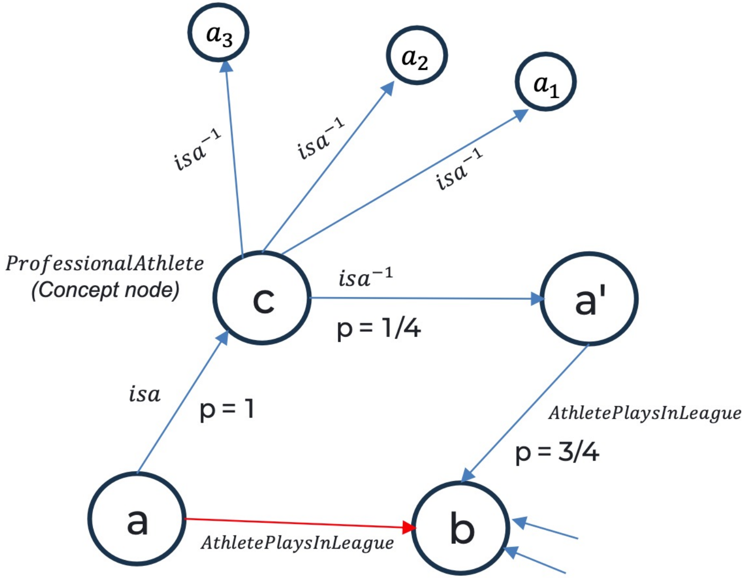
...

PRA

- Random Walk

- Start at a query node a
- The walk will reach c with probability 1
- If A is the set of ProfessionalAthlete in the KB, after two steps, the walk will have probability 1/|A| of being at any a' ∈ A
- after three steps, the walk will have probability |Aℓ|/|A| of being at any point b ∈ L

➔ **the ranking associated with this path** gives the **prior probability** of a **value b** being an athletic league for **a**



$$isa(a,c) \wedge isa^{-1}(c,a')$$
$$\wedge \quad AthletePlaysInLeague(a',b)$$
$$\Rightarrow \quad AthletePlaysInLeague(a,b)$$

PRA

- Path Ranking Algorithm

❖ Path Ranking

- Path $\boldsymbol{P}$ is defined as a sequence of relations $R_1, \dots, R_n$

$$T_0 \xrightarrow{R_1} \dots \xrightarrow{R_\ell} T_\ell$$

- Type of node can be end node of current relation and also can be start node of next relation

✓ $T_i = \mathrm{Range}(\mathrm{R_i}) = \mathrm{Domain}\left(\mathrm{R_{(i+1)}}\right)$

✓ $\mathrm{Domain(P)} \equiv T_0\,,\ range(P) \equiv T_i$

- One type of node " concept" used in experiment

➢ Can be connected different types of relations

$$P_1 : \quad concept \xrightarrow{AtheletePlayesForTeam} concept$$

$$P_2 : \quad concept \xrightarrow{AtheletePlayesForTeam} concept \xrightarrow{TeamPlaysInLeagure} concept$$

PRA

- Path Ranking Algorithm

- If P = $R_1, \ldots, R_l$ is empty path,

$$h_{s,P}(e) = \begin{cases} 1, & \text{if } e = s \\ 0, & \text{otherwise} \end{cases}$$

➢ Function h means probability distribution about ***node s*** to ***node e*** with ***path P***

✓ e: reaching node, s: seed node $\left(s \in domain(P)\right)$, l: length of path p

- If P = $R_1, \ldots, R_l$ is nonempty path, let $P' = R_1, \ldots, R_{l-1}$

$$h_{s,P}(e) = \sum_{e' \in range(P')} h_{s,P'}(e') \cdot P(e|e'; R_\ell)$$

- $P(e|e'; R_\ell) = \frac{R_\ell(e',e)}{|R_\ell(e',\cdot)|}$: the probability of e from e' with a one step random walk with edge type R_l

PRA

- Path Ranking Algorithm

- There can be some path between node s and node e, rank nodes by a linear model

$$\theta_1 h_{s,P_1}(e) + \theta_2 h_{s,P_2}(e) + \ldots \theta_n h_{s,P_n}(e)$$

✓ θ_n: *each path's appropriate weight*

- Then we can get score between node s and e by scoring function

$$score(e; s) = \sum_{P \in \mathbf{P}_\ell} h_{s,P}(e)\theta_P$$

PRA

- Path Ranking Algorithm

❖ Objective Function

$$o_i(\theta) = w_i[y_i \ln p_i + (1 - y_i)\ln(1 - p_i)],$$

- With Dataset $D = \{(x_i, y_i)\}$, $x_i = [\mathrm{h}_{\mathrm{s_i,P_1}}(\mathrm{t_i}), h_{s_i,P_2}(t_i), h_{s_i,P_3}(t_i), .P(y|\mathbf{x};\theta)$, $y_i = 0\ or\ 1$
- Log regression model to predict conditional probability

✓ $p(y_i = 1\ |x_i;\theta) = \frac{\{exp(\theta^T x_i)\}}{\{1+exp(\theta^T x_i)\}}$, w_i: $importance\ weight\ to\ each\ example$

$$O(\theta) = \sum_i o_i(\theta) - \lambda_1|\theta|_1 - \lambda_2|\theta|_2$$

- Maximize Objective function with L1-regularization, L2-regularization

PRA

- Data-driven path finding

Table 1: Number of paths in PRA models of maximum path length 3 and 4. Averaged over 96 tasks.

	ℓ=3	ℓ=4
all paths up to length L	15,376	1,906,624
+query support$\geq \alpha = 0.01$	522	5016
+ever reach a target entity	136	792
+L_1 regularization	63	271

- It is impractical to enumerate all possible relation path

- Any path node created during path finding needs to be **appeared by at least fraction α**
- must **retrieve** at least one **target entity t** in the training set
- Using **L1 regularization** (make path's weight by 0)

PRA

- Data-driven path finding

Table 1: Number of paths in PRA models of maximum path length 3 and 4. Averaged over 96 tasks.

	ℓ=3	ℓ=4
all paths up to length L	15, 376	1, 906, 624
+query support$\geq \alpha = 0.01$	522	5016
+ever reach a target entity	136	792
+L_1 regularization	63	271

- It is impractical to enumerate all possible relation path

- Any path node created during path finding needs to be **appeared by at least fraction α**
- must **retrieve** at least one **target entity t** in the training set
- Using **L1 regularization** (make path's weight by 0)

Conclusion

Previous work

In NELL, the method of learning the horn cause rule through N-FOIL and filling KB
was not able to make reliable inference because of considering one rule (path)

PRA

Statistic-based random walks can be used to consider various paths at the same time, learn the weights for each path, and infer based on them, allowing more flexible and reliable inference