Automated Knowledge Base Completion Using Collaborative Filtering and Deep Reinforcement Learning

Alisher Tortay, Jee Hang Lee, Chang Hwa Lee, Sang Wan Lee

한양대학교 인공지능연구실

INTRODUCTION : Automated Knowledge Base Completion

❖ 제안 논문 등장배경

AI application, 정보 검색 등에서 knowledge는 중요한 역할을 하는데 지식의 양이 많아질수록 missing, broken Links로 인한 incorrect information 문제로 항상 고통 받고 있음

❖ 제안 논문 목표

본 논문은 adding missing knowledge to the graph(Completion)에 중점(Not Error Detection)
(Error Detection이란, identifying wrong information in the graph)

Fig. 1. Example: Nodes are entities and relation are directed edges. The red edges represent missing relation between entities.

INTRODUCTION : Automated Knowledge Base Completion

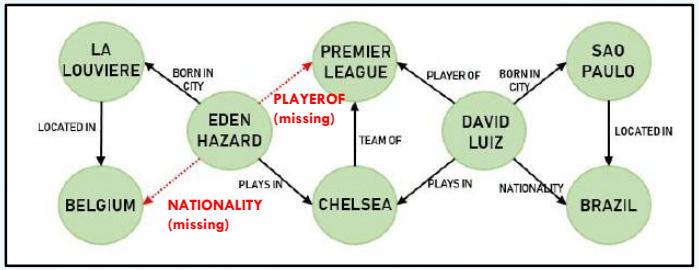
❖ 용어정리

: Relation(Reinforcement Learning에선 Action)

: Entity(Reinforcement Learning에선 State)

제안 논문 접근단계(1) : Collaborative Filtering

- Collaborative Filtering(1)
  - Missing Triplets(Entity-Relation-Entity)을 찾을 수 있음



지식 관계 그래프

Relation

Entity	LOCATEDIN	BORNIN	PLAYIN	TEAMOF	NATIONALITY	PLAYEROF
LALOUVIERE	1	0	0	0	0	0
BELGIUM	0	0	0	0	0	0
EDEN	0	1	1	0	0	0
PREMIER	0	0	0	0	0	0
CHELSEA	0	0	0	1	0	0
DAVID	0	1	1	0	1	1
SAO	1	0	0	0	0	0
BRAZIL	0	0	0	0	0	0

Collaborative Filtering을 사용하기 위해 만들어진 entity and relation

제안 논문 접근단계(1) : Collaborative Filtering

❖ Collaborative Filtering(2) : 누락된 Relation 찾는 방법

- Entity간 유사성 판단을 위해 Pearson correlation 사용

*Pearson correlation
두 변수 간의 관련성을 구하기 위해 보편적으로 사용되는 방법으로,
두 변수가 완전히 동일하면 +1, 전혀 다르면 0

Relation

Entity	LOCATEDIN	BORNIN	PLAYIN	TEAMOF	NATIONALITY	PLAYEROF
LALOUVIERE	1	0	0	0	0	0
BELGIUM	0	0	0	0	0	0
EDEN	0	1	1	0	0	0
PREMIER	0	0	0	0	0	0
CHELSEA	0	0	0	1	0	0
DAVID	0	1	1	0	1	1
SAO	1	0	0	0	0	0
BRAZIL	0	0	0	0	0	0

Collaborative Filtering을 사용하기 위해 만들어진 entity and relation

1. 벡터 공간에 Embedding된 Entity 사이에 Similarity 계산

$$Similarity(u,v) = \frac{(u-\bar{u})\cdot(v-\bar{v})}{||u-\bar{u}||_2||v-\bar{v}||_2}$$

2. Pearson correlation에서 적절한 t값을 설정하여, Entities 선택

$$S_u = \{v|Similarity(u,v) > t\}$$

3. 선택된 모든 Entities의 weighted average 계산하여 Entity 예측

$$Prediction(u) = \frac{\sum_{v\in S_u} v * Similarity(u,v)}{\sum_{v\in S_u} Similarity(u,v)}$$

4. 예측된 Entity의 Relation 개수 – 현재 비교하는 Entity의 Relation 개수의 차이로 누락된 Relation을 찾아냄

$$Gap(i,\ j) = Prediction(u)_j - u_j$$

제안 논문 접근단계(1) : Collaborative Filtering

❖ Collaborative Filtering(2) : 누락된 Relation 찾는 방법

- Entity간 유사성 판단을 위해 Pearson correlation 사용

*Pearson correlation
두 변수 간의 관련성을 구하기 위해 보편적으로 사용되는 방법으로,
두 변수가 완전히 동일하면 +1, 전혀 다르면 0

Relation

Entity	LOCATEDIN	BORNIN	PLAYIN	TEAMOF	NATIONALITY	PLAYEROF
LALOUVIERE	1	0	0	0	0	0
BELGIUM	0	0	0	0	0	0
EDEN	0	1	1	0	0	0
PREMIER	0	0	0	0	0	0
CHELSEA	0	0	0	1	0	0
DAVID	0	1	1	0	1	1
SAO	1	0	0	0	0	0
BRAZIL	0	0	0	0	0	0

Collaborative Filtering을 사용하기 위해 만들어진 entity and relation

1. 벡터 공간에 Embedding된 Entity 사이에 Similarity 계산

$$Similarity(u,v) = \frac{(u-\bar{u})\cdot(v-\bar{v})}{||u-\bar{u}||_2||v-\bar{v}||_2}$$

2. Pearson correlation에서 적절한 t값을 설정하여, Entities 선택

$$S_u = \{v|Similarity(u,v) > t\}$$

3. 선택된 모든 Entities의 weighted average 계산하여 Entity 예측

$$Prediction(u) = \frac{\sum_{v\in S_u} v * Similarity(u,v)}{\sum_{v\in S_u} Similarity(u,v)}$$

4. 예측된 Entity의 Relation 개수 – 현재 비교하는 Entity의 Relation 개수의 차이로 누락된 Relation을 찾아냄

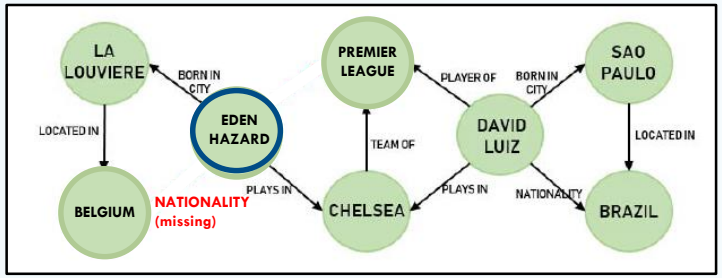
$$Gap(i,\ j) = Prediction(u)_j - u_j$$

제안 논문 접근단계(2) : Reinforcement Learning

❖ POLICY NETWORK

- 누락된(EDEN-NATIONALITY-?)의 ?해당 부분 찾아가는 과정
- (LA LOUVIERE-LOCATED IN) pair에 해당하는 다음 Entity를 선택하는 과정에서 LSTM 사용

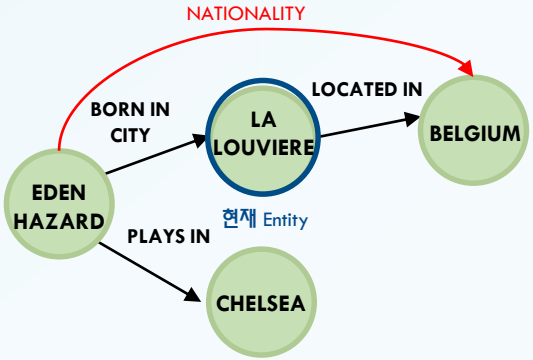
$$\mathbf{h}_t = \mathrm{LSTM}(\mathbf{h}_{t-1}, [\mathbf{a}_{t-1}; \mathbf{o}_t])$$



Relation이 누락된 지식 그래프

- 다음 행동(지식에선 Relation)의 확률을 구함

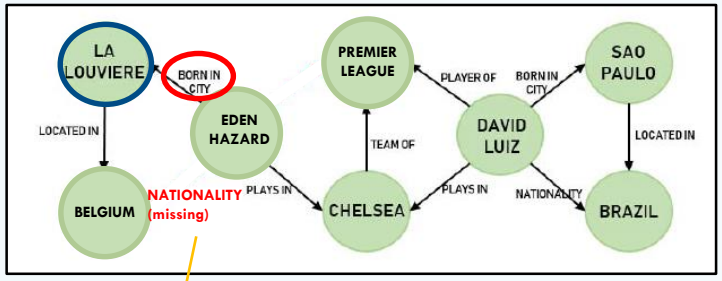
$$\mathbf{d}_t = \mathrm{softmax}(\mathbf{A}_t(\mathbf{W}_2\mathrm{ReLU}(\mathbf{W}_1[\mathbf{h}_t; \mathbf{o}_t; \mathbf{r}_q])))$$



제안 논문 접근단계(2) : Reinforcement Learning

❖ POLICY NETWORK

- 누락된(EDEN-NATIONALITY-?)의 ?해당 부분 찾아가는 과정
- (LA LOUVIERE-LOCATED IN) pair에 해당하는 다음 Entity를 선택하는 과정에서 LSTM 사용



Relation이 누락된 지식 그래프

$$\mathbf{h}_t = \mathrm{LSTM}(\mathbf{h}_{t-1}, [\mathbf{a}_{t-1}; \mathbf{o}_t])$$

이전 상태에서 선택한 Relation 현재 Entity

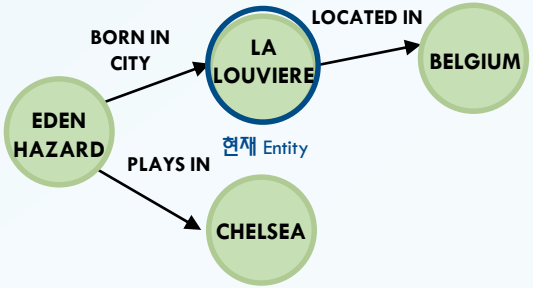
- 다음 행동(지식에선 Relation)의 확률을 구함

$$\mathbf{d}_t = \mathrm{softmax}(\mathbf{A}_t(\mathbf{W}_2\mathrm{ReLU}(\mathbf{W}_1[\mathbf{h}_t; \mathbf{o}_t; \mathbf{r}_q])))$$

모든 Relation 수만큼의 차원 벡터 중, LA LOUVIERE Entity에서 연결될 수 있는 Relation 벡터

음수에 대해서 0으로 처리

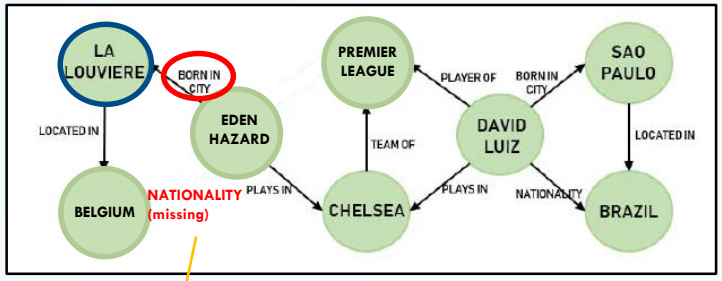
Collaborative Filtering을 통해 누락된 관계를 알게 된 (Entity-Relaion) pair로, 예제 그림에서 NATIONALITY에 해당



제안 논문 접근단계(2) : Reinforcement Learning

❖ POLICY NETWORK

- 누락된(EDEN-NATIONALITY-?)의 ?해당 부분 찾아가는 과정
- (LA LOUVIERE-LOCATED IN) pair에 해당하는 다음 Entity를 선택하는 과정에서 LSTM 사용



Relation이 누락된 지식 그래프

$$\mathbf{h}_t = \mathrm{LSTM}(\mathbf{h}_{t-1}, [\mathbf{a}_{t-1}; \mathbf{o}_t])$$

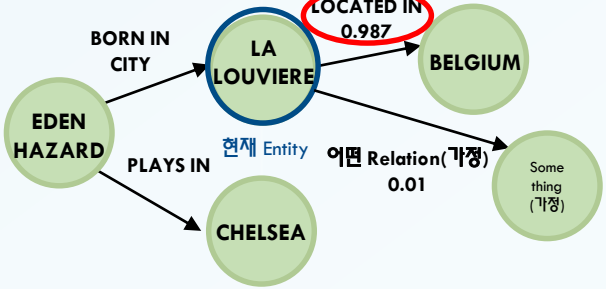
이전 상태에서 선택한 Relation 현재 Entity

- 다음 행동(지식에선 Relation)의 확률을 구함

$$\mathbf{d}_t = \mathrm{softmax}(\mathbf{A}_t(\mathbf{W}_2\mathrm{ReLU}(\mathbf{W}_1[\mathbf{h}_t; \mathbf{o}_t; \mathbf{r}_q])))$$

모든 Relation 수만큼의 차원 벡터 중,
LA LOUVIERE Entity에서 연결될 수 있는 Relation 벡터

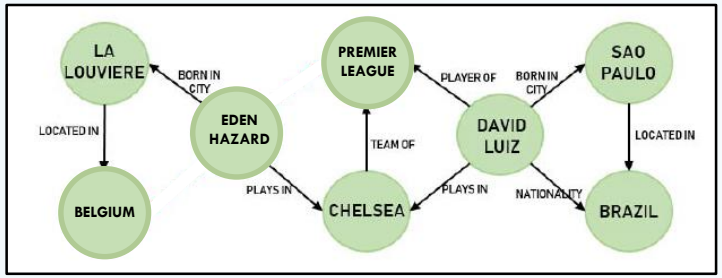
LA LOUVIERE에서 Relation 선택하게 되는 정책



제안 논문 접근단계(2) : Reinforcement Learning

❖ POLICY NETWORK

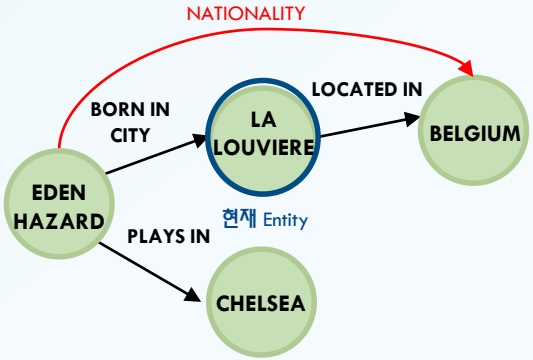
- 누락된(EDEN-NATIONALITY-?)의 ?해당 부분 찾아가는 과정
- (LA LOUVIERE-LOCATED IN) pair에 해당하는 다음 Entity를 선택하는 과정에서 LSTM 사용



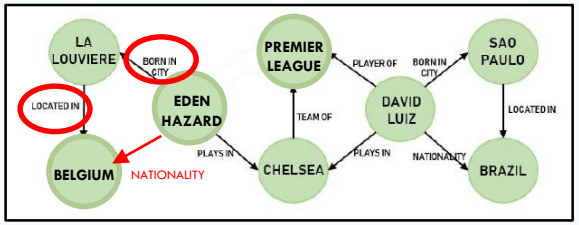
$$\mathbf{h}_t = \mathrm{LSTM}\left(\mathbf{h}_{t-1}, [\mathbf{a}_{t-1}; \mathbf{o}_t]\right)$$

- 다음 행동(지식에선 Relation)의 확률을 구함

$$\mathbf{d}_t = \mathrm{softmax}\left(\mathbf{A}_t(\mathbf{W}_2 \mathrm{ReLU}\left(\mathbf{W}_1 [\mathbf{h}_t; \mathbf{o}_t; \mathbf{r}_q]\right))\right)$$



(BORN IN-LOCARED IN)이 연속된다는 Rule based 방법을 통해서 NATIONALITY 지식 생성



제안 논문 접근단계(2) : Reinforcement Learning

❖ POLICY NETWORK TRAINING

$$J(\theta) = \mathbb{E}_{(e_1,r,e_2)\sim D}\mathbb{E}_{A_1,..,A_{T-1}\sim\pi_\theta}[R(S_T)|S_1 = (e_1,e_1,r,e_2)],$$

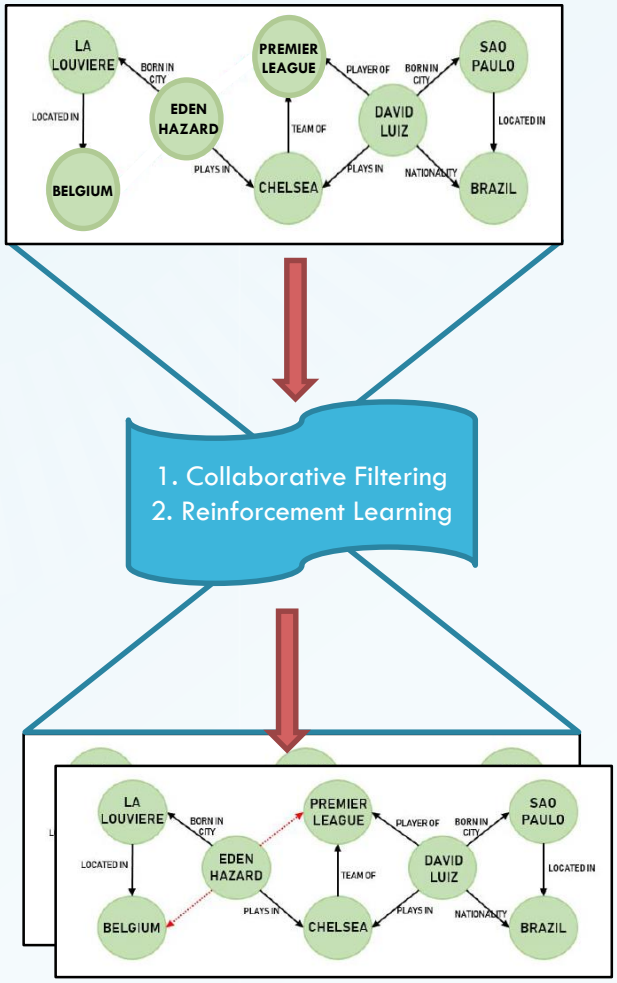
where we assume there is a true underlying distribution $(e_1, r, e_2) \sim D$. To solve this optimization problem, we employ REINFORCE (Williams, 1992) as follows:

θ : 기대할 수 있는 Reward의 값을 최대가 되도록 만드는 parameters

B. Triplet Completion Using a Deep RL Agent

For the second sub-problem, we basically follow a deep RL agent algorithm proposed in MINERVA [24], which infers the tail entity by addressing the task of path learning problem for KB completion described in Section II-B. In order to handle the partial observability of the MDP, MINERVA is designed as a randomized history-dependent policy and employ the function class expressed by long short-term memory network (LSTM) [33]. History is defined as a sequence of observations and actions taken. LSTM encodes the history as a continuous vector and the policy network chooses an action based on the history embedding, the head entity, and the relation. REINFORCE [34] is used to to train the RL agent.

단계 정리 및 장점



Input : 지식 베이스

Output : missing, broken link 감소하고 triplet이 증가한 지식 베이스

EXPERIMENTS AND EVALUATION

❖ *Data*

- *Unified Medical Language System(UMLS) :135 entities and 46 relations. The entities are concepts such as "enzyme", "mammal", and "virus". Most relations are verbs such as "measures", "occurs in", and "treats".*
- *Alyawarra Kinship : 104 entities and 25 relations. This dataset describes kinship relation between 104 members of Alyawarra tribe in Central Australia.*

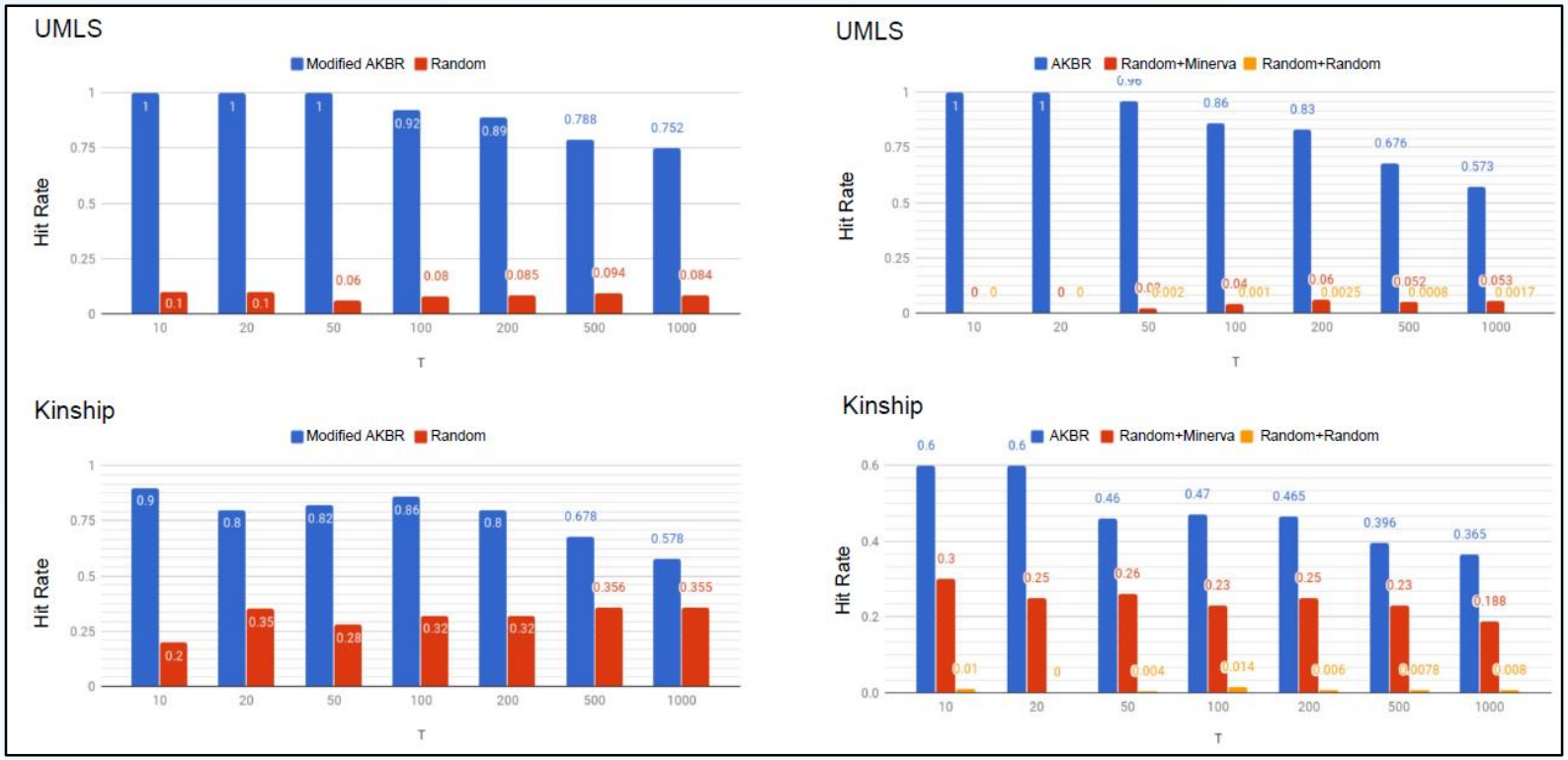
❖ *Evaluation Metric*

$$Precision = \frac{H}{T}$$

H : Completion결과, 맞은 개수
T : 임의의 수

- H 계산 방법은, test query triplets에서 마지막 Entity를 뺀 Entity-Relation 조합으로 Test query triplets 중 이 조합이 몇 개 나오는지

EXPERIMENTS AND EVALUATION



T : 지식 그래프 내에서 Entity-Relation-Entity 표현 triple으로, X축은 triplet 개수

(좌) Collaborative filtering을 사용한 Entity-Relation 추론 Precision
(우) Entity1-Relation-Entity2 triplet 추론 Precision

CONCLUSION

- 내부 지식 베이스 사용 외에 웹이나 데이터베이스 같은 외부 지식을 사용하여 Knowledge Completion 진행
- 시행 횟수(T)를 늘릴수록 성능이 떨어지는 현상 개선