

Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs

Intro.

- KGs는 missing relations를 추론하는 것이 가장 중요한 task 중 하나이다.
- 이를 흔히 knowledge base completion 또는 relation prediction이라고 한다. 이 tasks는 주어진 triple(h, r, t)이 valid 한지 그렇지 않은지에 대한 예측도 수반하고 있다.
여기서 h,r,t는 각각 head entity, relation, tail entity이다.
- 대표적인 methods는 knowledge embedding based models이 있다.
- 이는 크게 *translational models* 와 *CNN based models* 로 나뉜다.
- translational models는 간단한 operations와 limited parameters를 가지고 대체로 성능이 낮다.
- CNN 기반은 좀 더 복잡한 relations도 embedding이 가능하다.
- 하지만 두 방법 모두 triple independently 하게 진행이되고 그래서 주어진 entity 주변에 있는 풍부하고 잠재적인 relations를 "encapsulate" 하지 못 한다는 단점이 있다.
- 이 논문에서는 그래서 `generalized attention-based graph embedding for relation prediction` 을 제안한다.
- node classification에서 GATs는 node의 1-hop 이웃들에 대하여 가장 연관성이 높은 부분에 집중을 하는(가중치를 더 주는) 아이디어를 이용하고 있다.
- KG에서 relation prediction task를 수행함에 있어서, 이 논문이 제안하는 방법은 *generalizes and extends* 한 attention mechanism 을 보이고 있다. 이는 주어진 entity에서 multi-hop neighborhoods의 entity(node)와 relation(edge) features 모두에 attention을 적용한다.
 1. 주어진 노드 주변에 multi-hop relations를 수집한다.
 2. 다양한 관계속에서 node(entity)의 역할을 encapsulate(통합) 한다.
 3. 의미론적으로 유사한 관계 cluster에 존재하는 기존 지식을 통합한다.

to consolidate the existing knowledge present in semantically similar relation clusters
- 모델의 depth 가 증가함에 따라 떨어져있는 entites의 기여도는 지수적으로 감소한다. 이 문제를 해결하기 위해 저자는 `relation composition` (관계 구조(?))를 이용한다. 이는 auxiliary edge(보조 edge)를 n-hop neighbors에 추가해 주는 것으로, 좀 더 쉽게 entities 사이에 knowledge(information)이 흐를 수 있도록 해준다.
- 이 논문에서 제안하는 architecture는 encoder-decoder model이다.
encoder : generalized graph attention model
decoder : ConvKB
- 논문에서 주장하는 contributions
 1. 새로운 graph attention based embeddings를 학습하여 KG에서 relation prediction을 수행 함.
 2. 주어진 entity의 multi-hop neighborhood의 entity와 relation features에 대하여 graph attention을 수행 함.
 3. 다양한 dataset에서 모델을 평가함.

3 Our Approach

Background

- $G = (E, R)$, E 와 R 은 각각 entities(nodes) 집합과 relations(edges) 집합을 나타낸다.
- Triple (e_s, r, e_o) 는 G 에서 node e_s 와 e_o 사이의 관계 r 을 나타낸다.
- scoring function f 는 input triple $t=(e_s, r, e_o)$ 이 주어졌을때, valid triple이 존재할 likelihood 를 $f(t)$ 로 나타낸다.

Graph Attention Networks(GATs)

- GAT는 모든 node의 이웃들에 대하여 각각 어느정도의 중요도를 할당할지를 학습하게 된다.
- The input feature set of nodes to a layer is $\mathbf{x} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$
- A layer produces a transformed set of node feature vectors $\mathbf{x}' = \{\vec{x}'_1, \vec{x}'_2, \dots, \vec{x}'_N\}$
여기서 \vec{x}_i 와 \vec{x}'_i 는 각각 entity e_i 의 input 과 output embedding이다. 그리고 N 은 entities의 개수이다.
- A single GAT layer
$$e_{ij} = a(\mathbf{W}\vec{x}_i, \mathbf{W}\vec{x}_j)$$
 - 여기서 e_{ij} 는 edge (e_i, e_j) 의 attention value이다.
 - \mathbf{W} 는 linear transformation matrix이다.
 - $a(\cdot)$ 는 attention function 이다.
- attention value는 source node e_i 에 대하여 각각의 edge (e_i, e_j) features의 중요도를 나타낸다.
- 연결된 edge 들의 attention values set에 softmax를 취하여 α_{ij} (normalized attention coefficients) 를 구한다.

- GAT layer의 output

$$\vec{x}'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}\vec{x}_j\right)$$

- source node e_i 의 이웃 edges 들의 중요도를 가중치로 선형 변환된 이웃 노들의 hidden states($\mathbf{W}\vec{x}_j$) 에 weight sum을 계산한다 이후 activation function을 통과하고 최종 output을 구한다.
- The multihead attention (K attention heads)

$$\vec{x}'_i = \parallel_{k=1}^K \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \vec{x}_j\right)$$

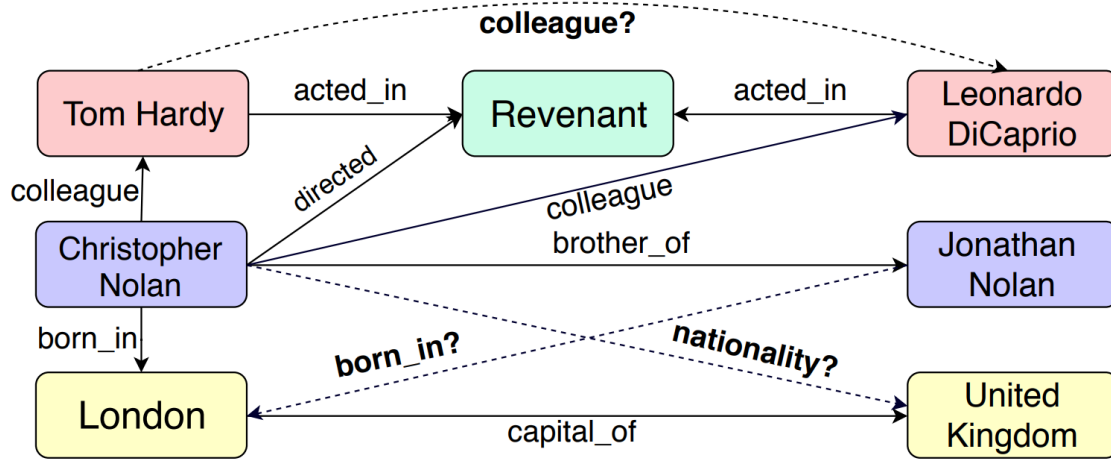
- \parallel 는 concatenation이다.
 - α_{ij}^k 는 k-th attention mechanism으로 계산된 normalized attention coefficients이다.
 - \mathbf{W}^k 는 k-th attention mechanism의 linear transformation matrix 이다.
- 마지막 layer의 output embedding 은 concatenate하지 않고 averaging 한다.

$$\vec{x}'_i = \sigma\left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \vec{x}_j\right)$$

Relations are Important

- GATs가 우수한 성능을 보이지만 KGs에 바로 적용되기엔 어려움이 있다. 왜냐하면 relation features를 반영하지 않아 불안정하기 때문이다.

- KGs에서 entites는 관련되 relation에 따라 다른 역할을 가지게된다. 예를 들어, Christopher Nolan은 brother 이자 director이다.



- 이러한 이유때문에 저자는 relation과 neighboring node features를 통합하여 embedding하는 attention mechanism을 제안한다.
- 각 layer들은 input으로 두 개의 embedding matrices를 받게된다.

1. Entity embeddings : $\mathbf{H} \in \mathbb{R}^{N_e \times T}$

- 이 matrix에서 i-th row 는 entity e_i 의 embedding 이다.
- N_e 는 전체 entities 개수이다.
- T 는 각 entity embedding의 feature dimensiond이다.

2. relation embeddings : $\mathbf{G} \in \mathbb{R}^{N_r \times P}$

- entity embedding 과 유사하다.

- 각 embedding matrices의 output은 아래와 같다.

$$\mathbf{H}' \in \mathbb{R}^{N_e \times T'}, \mathbf{G}' \in \mathbb{R}^{N_r \times P'}$$

- 임의의 triple 을 $t_{ij}^k = (e_i, r_k, e_j)$ 라 하자.
- attention mechanism

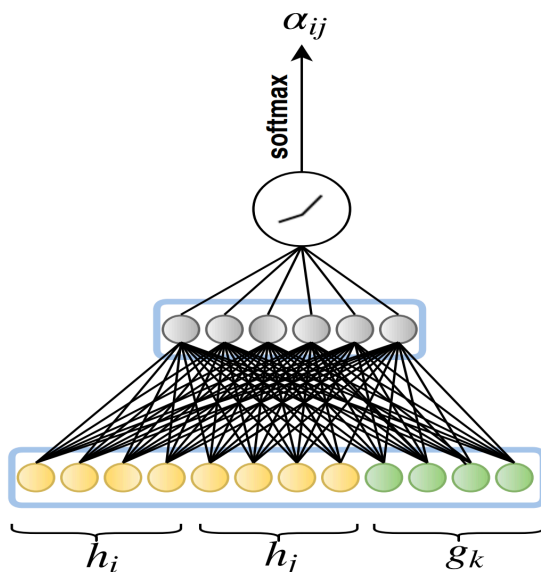


Figure 3: Attention Mechanism

- Triple t_{ij}^k 의 vector representation

$$\vec{c}_{ijk} = \mathbf{W}_1[\vec{h}_i \parallel \vec{h}_j \parallel \vec{g}_k]$$

- $\vec{h}_i, \vec{h}_j, \vec{g}_k$ 는 각각 entities와 relation의 embedding 이다.
- The importance of each triple t_{ij}^k 를 b_{ijk} 로 나타낸다.

$$b_{ijk} = \text{LeakyReLU}(\mathbf{W}_2 c_{ijk})$$

- source(head) entity와 연결된 tail entity와 relation에 대한 중요도를 계산한다.
- relative attention values α_{ijk} 를 구한다.

$$\alpha_{ijk} = \text{softmax}_{jk}(b_{ijk})$$

$$= \frac{\exp(b_{ijk})}{\sum_{n \in N_i} \sum_{r \in R_{in}} \exp(b_{inr})}$$

- N_i 는 entity e_i 의 이웃 entity 들이다.
- R_{ij} 는 entities e_i 와 e_j 사이에 연결되어 있는 relation set이다.
- entity e_i 의 새로운 embedding

$$\vec{h}'_i = \sigma\left(\sum_{j \in N_i} \sum_{k \in R_{ij}} \alpha_{ijk} \vec{c}_{ijk}\right)$$

- Multihead attention (M independent attention mechanisms)

$$\vec{h}'_i = \parallel_{m=1}^M \sigma\left(\sum_{j \in N_i} \alpha_{ijk}^m c_{ijk}^m\right)$$

- relation embedding matrix update

$$G' = G\mathbf{W}^R$$

- $\mathbf{W}^R \in \mathbb{R}^{T \times T'}$

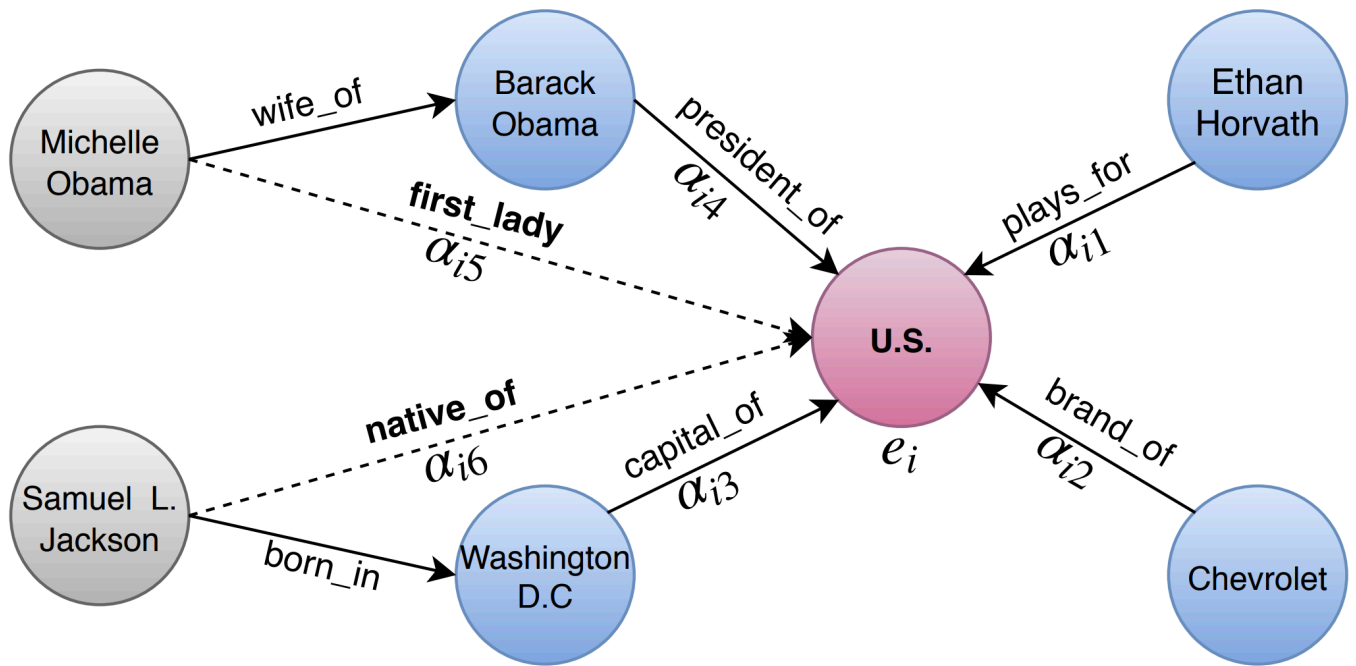
- The final embedding vectors

$$\vec{h}'_i = \sigma\left(\frac{1}{M} \sum_{m=1}^M \sum_{j \in N_i} \sum_{k \in R_{ij}} \alpha_{ijk}^m c_{ijk}^m\right)$$

- 새로운 임베딩을 학습하는 동안, entities는 그들의 초기 embedding information을 잃게 된다. 그래서 저자는 마지막 어텐션 레이어에서 얻어진 entity embedding 에 initail entity embedding information을 추가해줬다.

$$H'' = W^E H^t + H^f$$

- H^i 는 초기 entity embedding matrix이고 weight matrix $\mathbf{W}^E \in \mathbb{R}^{T^i \times T^f}$ 로 선형 변환하여 H^t 를 얻게된다.
- H^t 는 transformed entity embeddings이다.
- $H^f \in \mathbb{R}^{N_e \times T^f}$ 로 최종 어텐션 레이어로 구해진 embedding matrix이다.
- n-hop neighbors 떨어진 두 entity 사이에 대하여 auxiliary relation을 도입함으로써 edge의 개념을 directed path로 확장 시켰다. 이 보조 관계(auxiliary relation)의 embedding은 이 path까지의 모든 relations의 embeddings의 합으로 표현한다.



- 이 모델은 entity에서 떨어져있는 neighbors로 부터 정보를 반복적으로 축적해나간다. 즉 n layer model에서는 새로운 정보는 n-hop 이웃 전반에 걸쳐 축적되어 진다.

Training Objective

- 주어진 valid Triple $t_{ij}^k = (e_i, r_k, e_j)$ 에서 $\vec{h}_i + \vec{g}_k \approx \vec{h}_j$ 가 될 수 있도록 embedding 하는 것이 목표가 된다.
- idea는 e_i 랑 reation r_k 로 연결된 entity 중 가장 가까운 entity가 e_j 여야 하는 것이다.
- Hinge-loss를 써서 다음과 같이 loss function을 나타낼 수 있다.

$$L(\Omega) = \sum_{t_{ij} \in S} \sum_{t'_{ij} \in S'} \max\{d_{t'_{ij}} - d_{t_{ij}} + \gamma, 0\}$$

- $\gamma > 0$ 은 margin hyper-parameter이다.
- S 는 valid triples set 이다. S' 는 invalid triples set으로 다음과 같이 주어진다.

$$S' = \{t_{i'j}^k | e'_j \in E \setminus e_j\} \cup \{t_{ij}^k | e'_j \in E \setminus e_j\}$$

- e_i 가 아닌 entity가 head이고 e_j 와 r_k 의 reation으로 연결되어 있는 triple set + e_j 가 아닌 entity가 tail 이고 head가 e_i 로 r_k 의 관계로 연결되어 있는 triple set

4 Experiments and Results

Evaluation Protocol

- Relational prediction은 미완성된 triple에서 head나 tail entity를 예측하는 것입니다. 즉 (e_i, r_k, e_j) 에서 한 entity가 살아진 (r_k, e_j) 또는 (e_i, r_k) 인 경우입니다.
- valid triple (e_i, r_k, e_j) 에 대하여 N-1개의 corrupt triples를 구합니다. 이때 e_i 대신에 $e'_i \in E \setminus e_i$ 가 들어갑니다. 모델은 corrupt triples의 scores를 계산하고 valid triple score와 함께 rank를 구합니다.
- MR은 true triple 의 score 순위 평균으로 낮을 수록 좋습니다.
- MRR은 corrupt triples 사이에 true triple이 얼마나 상위에 랭크되어 있는가를 수치화한 것으로 클 수록 좋습니다.
- Hits#N 은 N개의 상위 rank에 ture triple이 있는지를 나타내며 전체 true triples set에 대하여 평균으로 구합니다. 높

을 수록 좋습니다.

	WN18RR					FB15K-237				
	MR	MRR	Hits@N			MR	MRR	Hits@N		
			@1	@3	@10			@1	@3	@10
DistMult (Yang et al., 2015)	7000	0.444	<u>41.2</u>	<u>47</u>	50.4	512	0.281	19.9	30.1	44.6
ComplEx (Trouillon et al., 2016)	7882	<u>0.449</u>	40.9	46.9	53	546	0.278	19.4	29.7	45
ConvE (Dettmers et al., 2018)	4464	0.456	41.9	47	53.1	245	<u>0.312</u>	<u>22.5</u>	34.1	<u>49.7</u>
TransE (Bordes et al., 2013)	2300	0.243	4.27	44.1	53.2	323	0.279	19.8	<u>37.6</u>	44.1
ConvKB (Nguyen et al., 2018)	1295	0.265	5.82	44.5	<u>55.8</u>	<u>216</u>	0.289	19.8	32.4	47.1
R-GCN (Schlichtkrull et al., 2018)	6700	0.123	20.7	13.7	8	600	0.164	10	18.1	30
Our work	<u>1940</u>	0.440	36.1	48.3	58.1	210	0.518	46	54	62.6

Table 2: Experimental results on WN18RR and FB15K-237 test sets. Hits@N values are in percentage. The best score is in **bold** and second best score is underlined.

	NELL-995					Kinship				
	MR	MRR	Hits@N			MR	MRR	Hits@N		
			@1	@3	@10			@1	@3	@10
DistMult (Yang et al., 2015)	4213	0.485	40.1	52.4	61	5.26	0.516	36.7	58.1	86.7
ComplEx (Trouillon et al., 2016)	4600	0.482	39.9	52.8	60.6	2.48	0.823	73.3	89.9	97.11
ConvE (Dettmers et al., 2018)	3560	<u>0.491</u>	<u>40.3</u>	<u>53.1</u>	<u>61.3</u>	<u>2.03</u>	<u>0.833</u>	<u>73.8</u>	<u>91.7</u>	98.14
TransE (Bordes et al., 2013)	2100	0.401	34.4	47.2	50.1	6.8	0.309	0.9	64.3	84.1
ConvKB (Nguyen et al., 2018)	600	0.43	37.0	47	54.5	3.3	0.614	43.62	75.5	95.3
R-GCN (Schlichtkrull et al., 2018)	7600	0.12	8.2	12.6	18.8	25.92	0.109	3	8.8	23.9
Our work	<u>965</u>	0.530	44.7	56.4	69.5	1.94	0.904	85.9	94.1	<u>98</u>

Table 3: Experimental results on NELL-995 and Kinship test sets. Hits@N values are in percentage. The best score is in **bold** and second best score is underlined.