

# Relational Message Passing for Knowledge Graph Completion

## Abs.

- Knowledge graph completion은 KG에서 entities 사이의 missing relations를 예측하는 것을 목표로 한다.
- 이 논문에서는 *relational message passing* methods를 제안하여 KG completion을 수행하였다.
- 기존의 embedding-based methods들과 다르게, relational message passing은 edge features(i.e., relation types)만 고려하면 된다는 특징이 있다.
- 그리고 edges 사이에서 relational message를 반복적으로 전달하며, 이웃 정보를 aggregate 하게된다.
- relational message passing frameworks로 주어진 entity pair에 대한 두 종류의 neighborhood topology를 모델링 할 수 있다.
  1. Relational context : 주어진 entity pair에 인접한 edges의 relation types를 잡아낸다.
  2. Relational paths : KG에서 주어진 두 entities 사이의 relative position을 나타낸다.

## Intro.

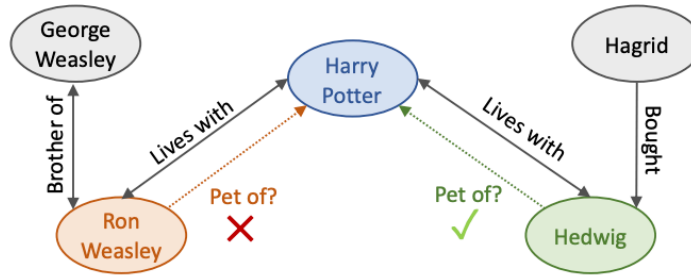
- KG 는 대개 triplets의 모음으로 구성되어 있다. 각 triplet은 (h,r,t)로 head entity h는 relational type r 로 tail entity t에 연결되어 있다.
- KG는 imcomplete 하고 noisy하다. 이런 이슈를 다루기 위해서 다양한 KG completion methods 가 소개되었다.
- 일반적으로 relation types는 고르게 분포되어 있지 않다. 하지만 서로 공간적으로 연관되어 있다.("spatially correlated with each other"). 예를 들어 "graduated from" 이라는 relation은 "person.birthplace" 와 "university.location"에 더 적합하다. "movie.language"와는 어색하다.
- 그러므로, 주어진 entity pair (h,t)에서 h와 t의 이웃 links의 relation types를 특징짓는 것은 h와 t 사이의 relation type을 추론할때 유의미한 정보를 제공한다.
- 저자는 주어진 entity pair의 neighborhood structure를 잘 잡아내는 *message passing* 을 사용하려고 했는데, 기존의 message passing methods는 보통 메시지가 노드와 연관되어 있고 노드에서 노드로 반복적으로 전달되게 된다.
- 이는 edge features가 더 중요한 KG에 적합하지 않다.

## Relational message passing

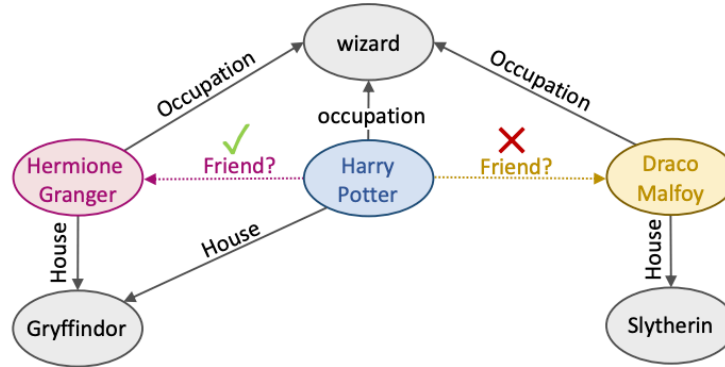
- 기존의 message passing 과 다르게 relational message passing은 오직 edge features만 고려한다. 그리고 edge 의 메시지를 곧 바로 이웃 edges에 전달한다.
- 이 방법은 기존의 KG embedding methods 보다 세 가지 이점을 가지고 있다.
  1. inductive : 추론단계에서 training data에서 나타나지 않았던 entities를 다룰 수 있다.
  2. Storage-efficient : entities의 embeddings를 계산하지 않는다.
  3. explainable : relation types 사이의 correlation strength를 modeling함으로써 예측 결과에 대한 설명가능성을 제공한다.
- relational message passing이 가지는 잠재적인 이슈는 computational complexity가 node-based message passing에 비해 매우 커질 수 있다는 것이다.
- 이 문제를 풀기위해 **alternate relational message passing** 을 제안한다. 이는 relational messages를 nodes와 edges에 번갈아가며 전달하는 방법이다. 이 방법으로 높은 시간 효율과 기존의 node-based message passing과 같은 계산 복잡도를 보였다.

## Relational context and relational paths

- alternate relational message passing framework로 주어진 entity pair (h,t)에 대하여 다음 두 종류의 local subgraph topology를 연구하였다.



(a) Consider we aim to predict whether Ron Weasley or Hedwig is a Pet of Harry Potter. Both entities have the same relational path (Lives with) to Harry Potter but they have distinct relational context: Ron Weasley has {Brother of, Lives with}, while Hedwig has {Bought, Lives with}. Capturing the relational context of entities allows our model to make a distinction between Ron Weasley, who is a person, and Hedwig, which is an owl.



(b) Two head entities Hermione Granger and Draco Malfoy have the same relational context {Occupation, House}, but different relational paths to the tail entity Harry Potter {(House, House), (Occupation, Occupation)} vs. {(Occupation, Occupation)}, which allows our model to predict friendship between Harry Potter and Hermione Granger vs. Draco Malfoy.

**Figure 1: (a) Relational context of an entity and (b) relational paths between entities. Our model is able to capture both.**

1. *Relational context* : 주어진 entity의 neighboring relations를 잡아내는 것은 중요하다. 왜냐하면 neighboring relations는 주어진 entity의 nature나 "type"에 대한 중요한 정보를 제공하기 때문이다.

Many entities in KGs are not typed or are very loosely typed

(entities들의 정보가 완성되어 있지 않다는 뜻으로 이해를 하였습니다.)

저자는 multi-hop neighboring endges를 aggregate 하기 위해 multi-layer relational message passing scheme을 구현하였다.

(위 그림 (a)는 해리포터는 론과 헤드위그랑 같이 살고 있는데 누가 해리의 반려동물인지 추론하는 문제입니다. 여기서 "Lives with"이라는 관계는 두 entity 모두 같습니다. 하지만 론은 다른 edge의 type으로 "Brother of"를 가지고 있고 헤드위그는 "Bought" 를 가지고 있습니다. 즉 multi-hop neighboring edges를 이용하여 새로운 ration을

추론할 수 있게됩니다.")

2. Relational paths : relational context만 사용하게 되면 relative position 을 변별할 수 없다. (h,t) 사이의 relational paths 집합을 잡아내는 것 역시 중요하다. entities 사이 연결의 다양한 paths들은 그들 관계의 본질(본연의 성질)를 발견할 수 있고 또 예측에 도움이 된다. 따라서 h와 t 사이에 연결된 모든 relational paths를 모두 계산하고 이 paths를 따라서 relational message를 전달한다. 마지막으로 attention mechanism 을 사용하여 다양한 relational paths의 representations를 선택적으로 aggregate 한다. 그리고 마지막으로 이 두 방법을 결합하여 최종 relation 예측을 하게된다.

## Our Approach

### 3.1 Relational Message Passing Framework

#### Alternate relational message passing

$$\begin{aligned} m_v^i &= A_1(\{s_e^i\}_{e \in N(v)}), \\ m_e^i &= A_2(m_v^i, m_u^i), \quad v, u \in N(e), \\ s_e^{i+1} &= U(s_e^i, m_e^i) \end{aligned}$$

- 각 노드 v에서, v에 연결된 edges( $e \in N(v)$ ) 들을 aggregate function  $A_1(\cdot)$  으로 aggregate한다.
- 그리고 edge e에 대해 양 끝에 연결된 nodes( $v, u \in N(e)$ )의 messages를 aggregation해서 message  $m_e^i$  를 얻을 수 있다. 최종적으로 edge의 hidden state  $s_e^i$  와  $m_e^i$  를 가지고 update를 수행한다.
- alternate relational message passing을 직관적으로 이해해보자면, 여기서 nodes는 "distribution centers"의 역할을 하는데, 이웃 edges들로 부터 message를 수집하고 일시적으로 저장하는 역할을 한다. 그리고 수집된 messages를 각 nodes의 이웃 edges에게 다시 전파하는 역할을 한다.