



16주차 세션

DL팀 이다현

Heterogeneous Graphs



Heterogeneous Graphs

✓ Heterogeneous Graph (다양한 종류로 이루어진 그래프) 의 정의

$$G = (V, E, R, T)$$

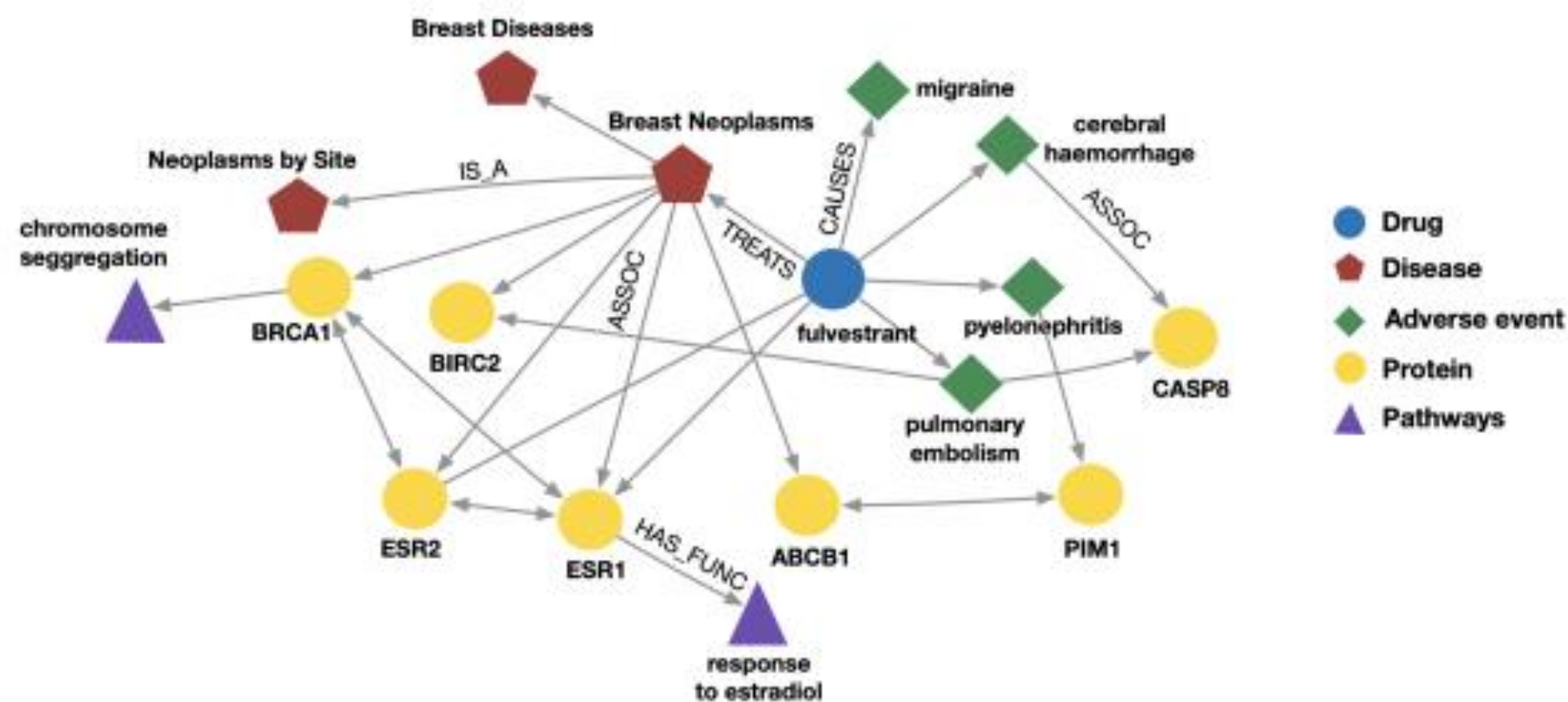
- Nodes with node types $v_i \in V$
- Edges with relation types $(v_i, r, v_j) \in E$
- Node type $T(v_i)$
- Relation type $r \in R$ 이 부분에 대해 자세히 볼 예정

엣지는 두 노드 사이에 어떤 연결종류 r 을 가지는지에 따라 결정되고 노드 종류는 이를 결정하는 함수에 각 노드가 입력되어 결정된다.

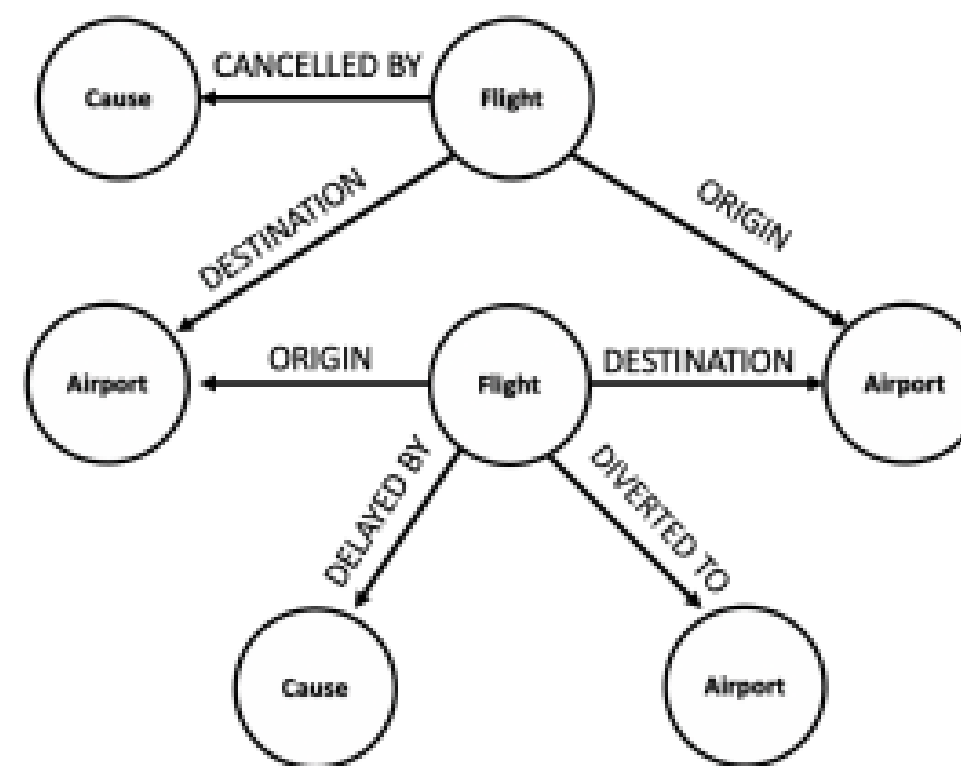
Heterogeneous Graphs

✓ Heterogeneous Graph 의 예시

복잡한 그래프는 다양한 종류의 엣지와 노드를 이용해 표현할 수 있다.



Biomedical Knowledge Graphs



Event graphs

- (약품, 질병, 단백질) 등의 노드
- (치료, 유발, 효과) 등의 엣지

- (공항, 비행, 경로) 등의 노드
- (연착, 출발, 목적지) 등의 엣지

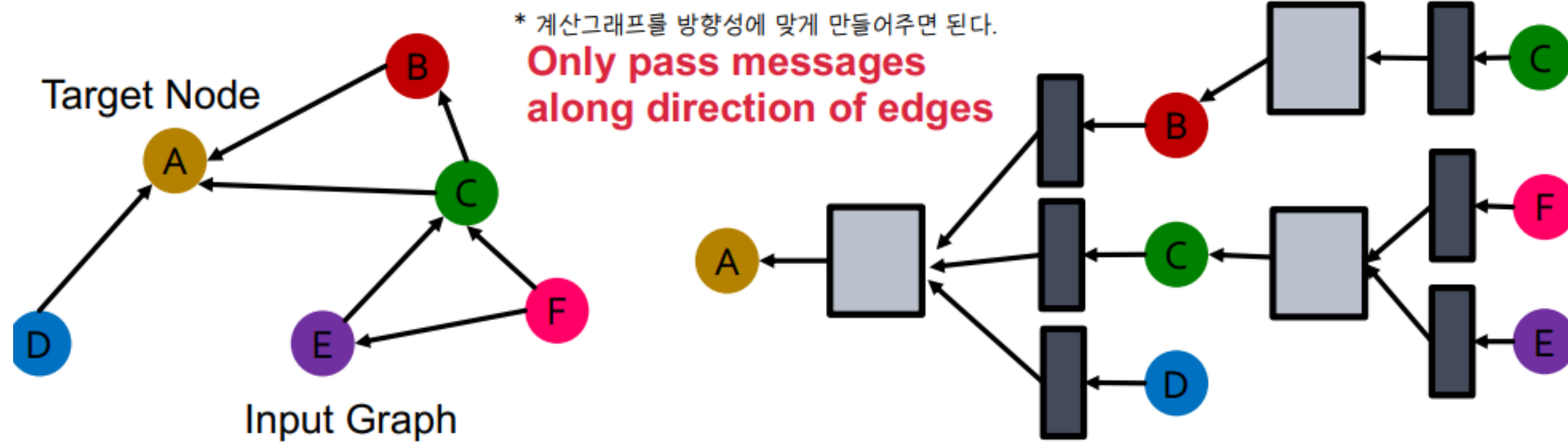
RGCN



RGCN

✓ Relational GCN

- 방향성 있는 (directed) 그래프에서 GCN 의 동작

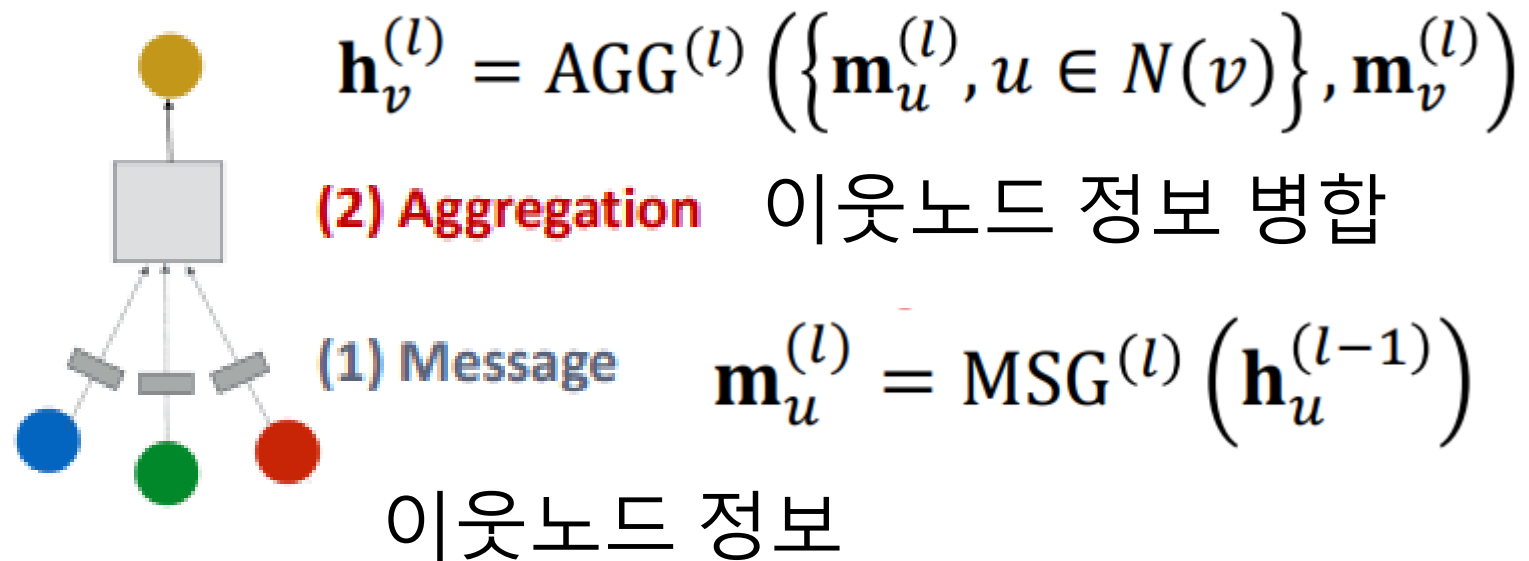


RGCN

✓ GCN 복습

- single GNN layer

= Message + Aggregation



Nonlinearity (activation):

활성화 함수에 통과시켜 비선형성 추가

Graph Convolutional Networks (GCN)

The diagram shows the GCN equation with various components annotated in Korean:

$$\mathbf{h}_v^{(l)} = \sigma \left(\mathbf{W}^{(l)} \sum_{u \in N(v)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|} \right)$$

Annotations:

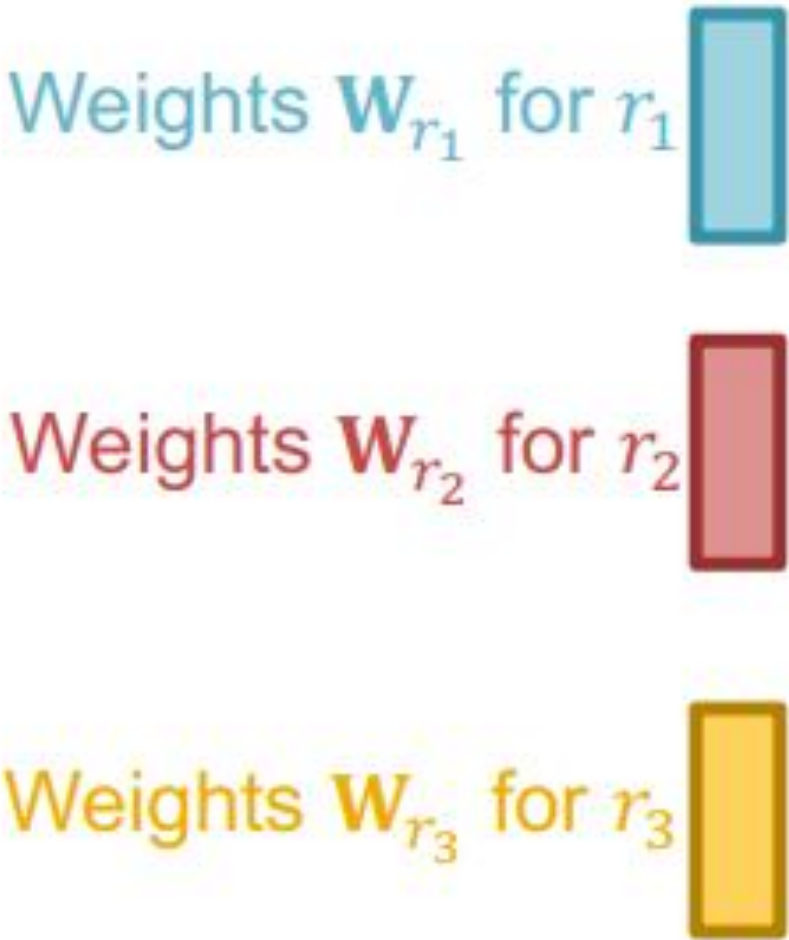
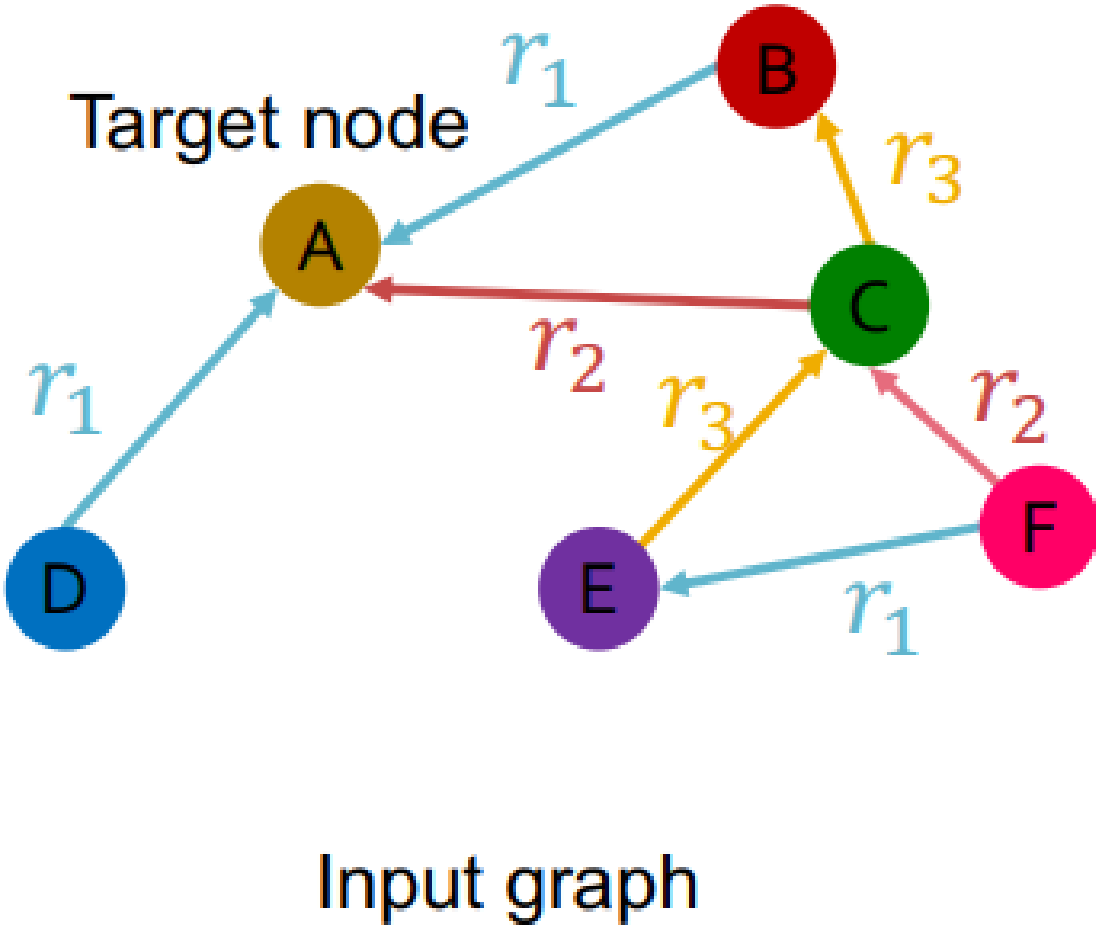
- 각 레이어마다 개별적인 가중치 행렬을 가짐 (Each layer has its own individual weight matrix)
- 각 레이어의 이웃노드의 벡터의 평균 (Average of neighbor node vectors of each layer)
- * 1 번째 이전 레이어의 이웃노드 u 의 벡터 (Vector of neighbor node u of the previous layer)
- * 상수값 : v 노드의 이웃노드의 수 (Constant value: number of neighbor nodes of node v)
- * 1 번째 레이어의 노드 v 의 벡터 (Vector of node v of the first layer)

- 노드 벡터 인코딩 layer
- W 를 통해 message passing 의 가중치를 부여함 (선형변환)
- 임베딩 벡터의 크기를 유지시켜주기 위해 이웃노드의 수 $N(v)$ 로 정규화

RGCN

✓ Relational GCN

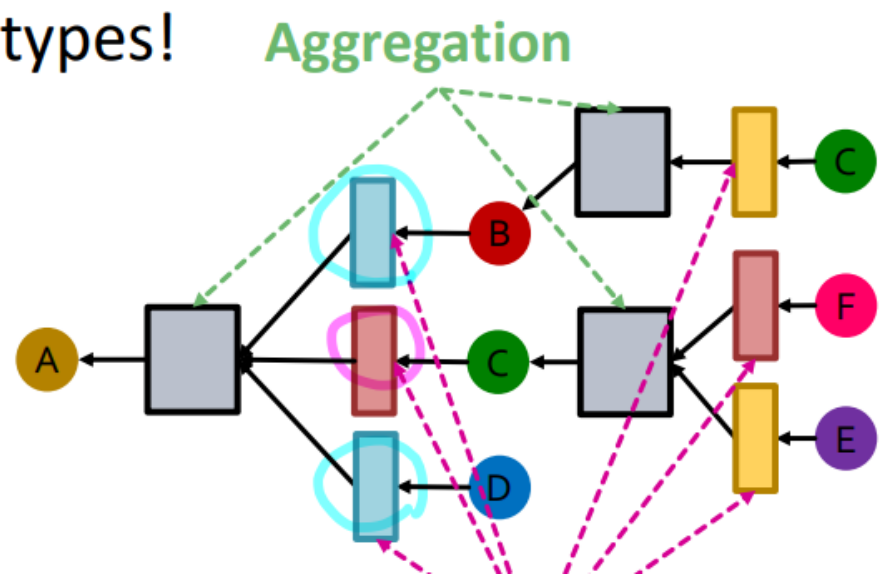
- 여러 개의 relation type 을 가진 그래프는 어떻게 인코딩할까?



- relation 종류에 따라 다른 weight matrix W 를 가질 수 있도록 한다.

Message

$$W^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|}$$



RGCN

✓ Relational GCN

$$\mathbf{h}_v^{(l+1)} = \sigma \left(\sum_{r \in R} \sum_{u \in N_v^r} \frac{1}{c_{v,r}} \mathbf{W}_r^{(l)} \mathbf{h}_u^{(l)} + \mathbf{W}_0^{(l)} \mathbf{h}_v^{(l)} \right)$$

* 매 layer 마다,
매 r 마다 다른 가중치
행렬을 가지게 됨
(많은 파라미터 수)

- 각 relation r 마다 합계가 이루어진다.
- 가중치 행렬 W 는 각 relation type 마다, 레이어마다 가지게 된다.

① 이웃노드의 임베딩 벡터

$$\mathbf{m}_{u,r}^{(l)} = \frac{1}{c_{v,r}} \mathbf{W}_r^{(l)} \mathbf{h}_u^{(l)}$$

Normalized by node degree
of the relation $c_{v,r} = |N_v^r|$

- * 메시지는 **동일한 r**을 가지는 이웃노드의 이전 레이어의 선형벡터 변환을 통해 표현된다.
- * r을 가지는 이웃노드의 수 c 로 정규화 된다.

② 자기 자신의 임베딩 벡터

$$\mathbf{m}_v^{(l)} = \mathbf{W}_0^{(l)} \mathbf{h}_v^{(l)}$$

해당 레이어의 가중치
행렬과의 연산으로 메시지가
생성된다.

③ Aggregation : sum

$$\mathbf{h}_v^{(l+1)} = \sigma \left(\text{Sum} \left(\left\{ \mathbf{m}_{u,r}^{(l)}, u \in \{N(v)\} \cup \{v\} \right\} \right) \right)$$

RGCN

✓ Relational GCN : Scalability

- RGCN 은 여러 관계를 가지는 그래프에 대한 모델링이지만, 파라미터 수가 너무 많아 과적합 발생 위험

$$\mathbf{W}_r^{(1)}, \mathbf{W}_r^{(2)} \dots \mathbf{W}_r^{(L)}$$

↪ L개의 hidden layer 를 가지는 RGCN 이 가지는 각 r마다의 가중치 행렬

■ The size of each $\mathbf{W}_r^{(l)}$ is $d^{(l+1)} \times d^{(l)}$

* 하나의 가중치 행렬은 $d^{(l+1)} \times d^{(l)}$ 개의 파라미터를 가진다.

* $d^{(l)}$ 은 l 번째 레이어의 hidden (은닉) 차원

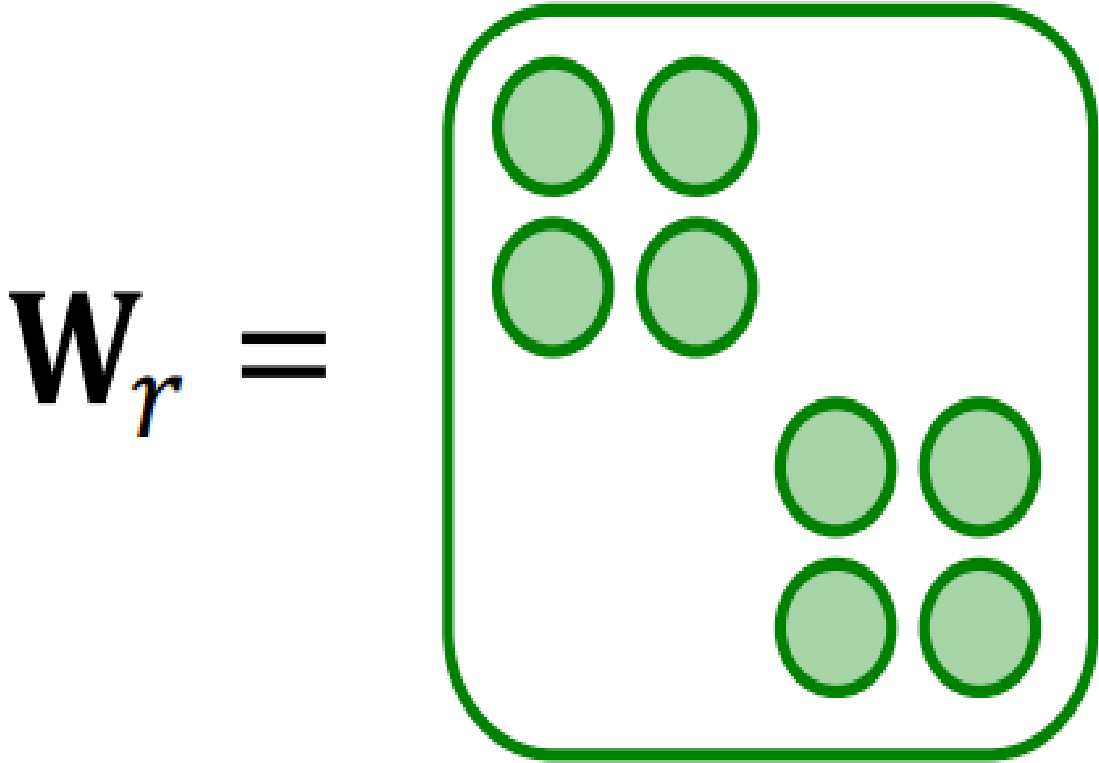
l 번째 layer 에서 r 의 관계성을 가지는
부분의 가중치 행렬

↪ 과적합 해결방법 : block diagonal matrices, Basis learning

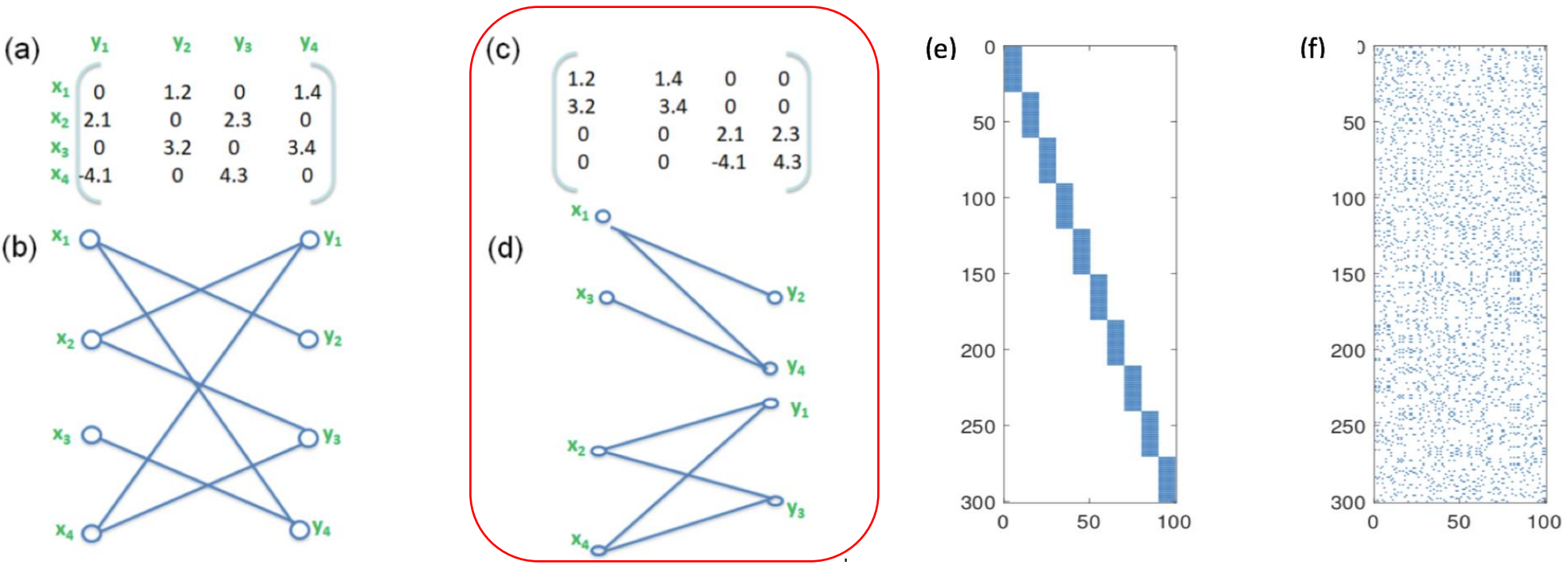
RGCN

✓ Relational GCN : Scalability – Solution (1) Block diagonal Matrices

(1) Block diagonal matrices



- 하나의 행렬에 대해 대각 방향으로만 부분 행렬이 존재하도록 만드는 것 → 행렬이 sparse 해져서 훨씬 파라미터 수가 줄어든다.



파라미터 개수가 오른쪽과 같이 훨씬 줄어들게 된다.

from $d^{(l+1)} \times d^{(l)}$ to $B \times \frac{d^{(l+1)}}{B} \times \frac{d^{(l)}}{B}$.

DNN 구조로 보면 근처 노드에 대해서만 연결되어 있는 모습
↳ 근처 노드만 봐서 모든 원소와의 관계 파악은 X

RGCN

✓ Relational GCN : Scalability – Solution (2) Basis learning

(2) Basis learning • 다른 relation 의 파라미터를 공유하도록 하는 방법

$$\mathbf{W}_r = \sum_{b=1}^B \mathbf{a}_{rb} \cdot \mathbf{V}_b,$$

↪ 모든 relation r 의 basis 행렬을 만들고, 이 기본 행렬의 선형결합으로 각 관계의 W 가중치를 표현한다.

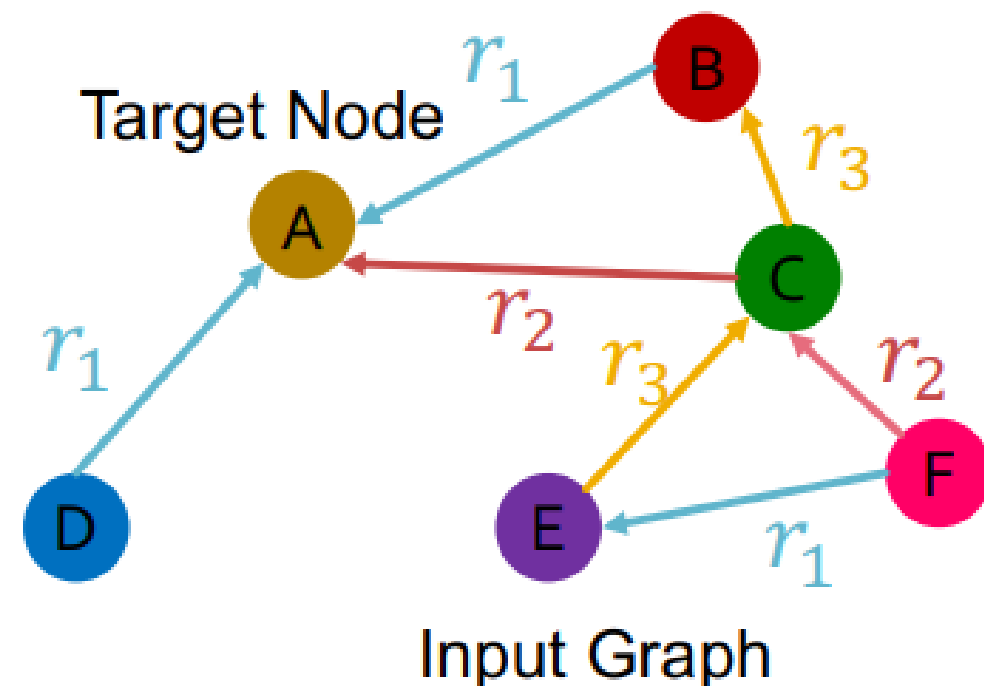
↪ α : 선형결합의 계수로, 학습되는 스칼라 값이다.

RGCN

✓ Relational GCN 예시 : Node classification

(1) 주어진 노드의 레이블을 예측하는 Task

- RGCN 은 분류하고자 하는 노드의 임베딩 벡터만을 사용함



$$\mathbf{h}_A^{(L)} \in \mathbb{R}^k,$$

RGCN

✓ Relational GCN 예시 : Link prediction

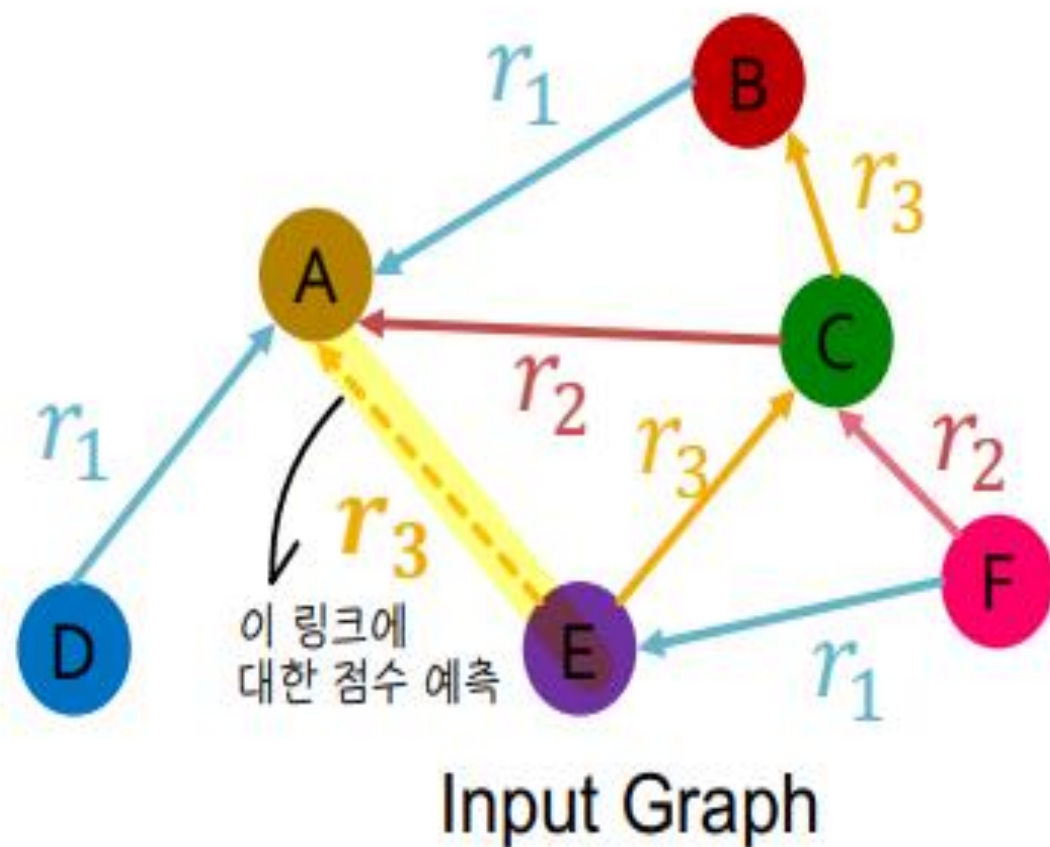
(2) 엣지가 존재할 가능성 예측

• 다양한 relation 이 존재하므로, 데이터 구성에 신경써야 한다.

• (E, r3, A) : Training supervision edge (예측할 엣지)

• 나머지 엣지 : Training message edges

→ RGCN 을 사용해 (E,r3,A) score 를 매기자 (엣지가 존재할 가능성)



✓ E와 A 의 임베딩 벡터를 얻는다.

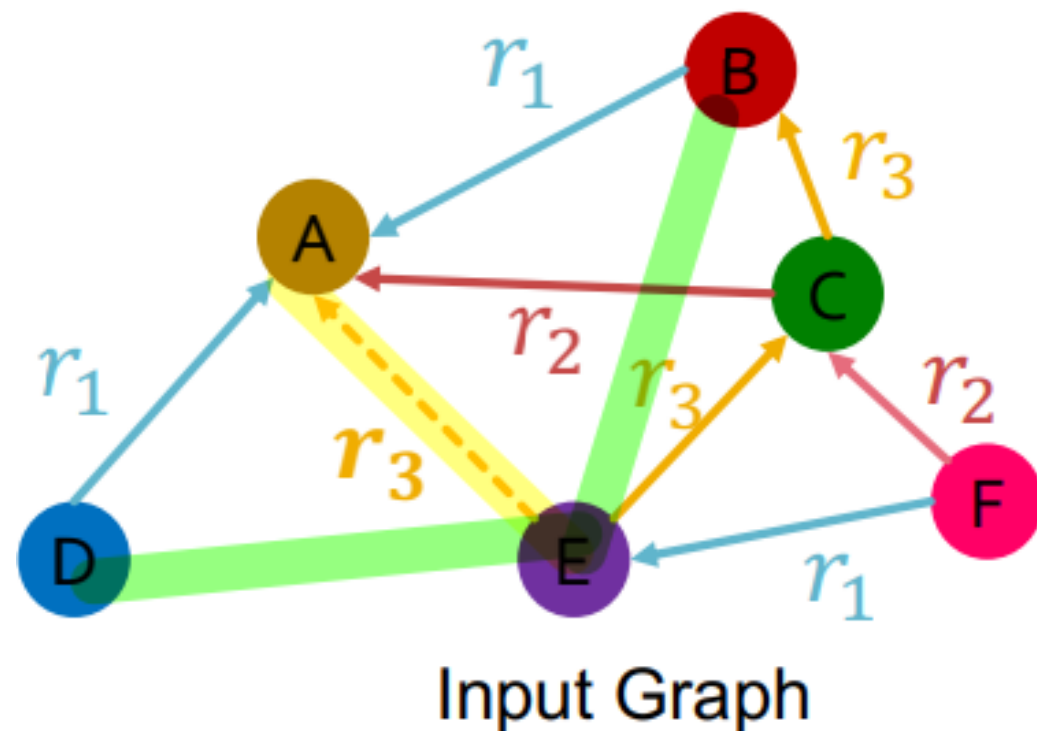
✓ 두 벡터에 대해 점수를 구할 함수를 정의한다.

$$f_{r_1}(\mathbf{h}_E, \mathbf{h}_A) = \mathbf{h}_E^T \mathbf{W}_{r_1} \mathbf{h}_A, \mathbf{W}_{r_1} \in \mathbb{R}^{d \times d}$$

RGCN

✓ Relational GCN 예시 : Link prediction

Training



1. RGCN 을 통해 구한 임베딩 벡터로 supervision edge (E,r3,A) 의 score 를 구한다.
2. Negative edge (초록선) 를 만든다. Supervision edge 에 포함되지 않는 조합을 negative edge 라 부름
3. RGCN 을 통해 구한 임베딩 벡터로 negative edge 에 대한 score 를 구한다.
4. Cross entropy 를 이용해 최적화

$$\ell = -\log \sigma \left(f_{r_3}(h_E, h_A) \right) - \log(1 - \sigma(f_{r_3}(h_E, h_B)))$$

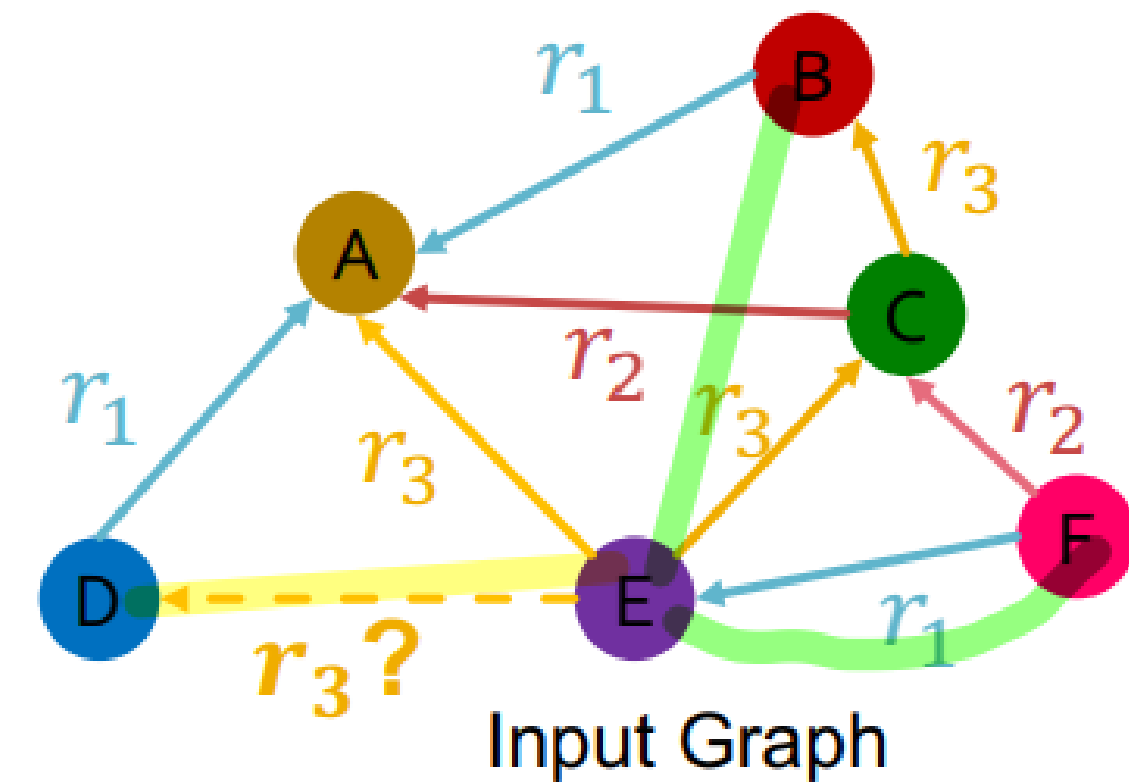
Sigmoid function

Training supervision edge 에 대한 score 는 최대화,
negative edge 에 대한 score 는 최소화 한다.

RGCN

✓ Relational GCN 예시 : Link prediction

Evaluation

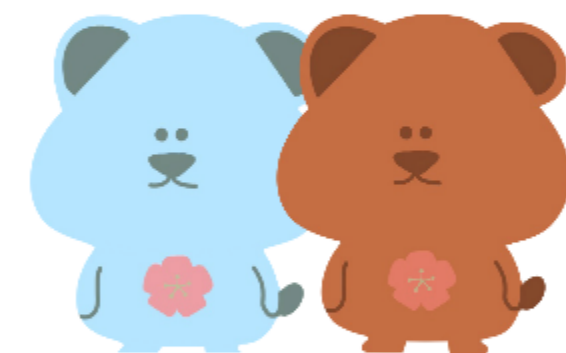


1. (E,r3,D) validation edge 에 대해 score 를 구한다.
2. Negative edge 에 대한 score 를 구한다.
3. Score 를 기준으로 (E,r3,D) 의 ranking 을 매긴다.
4. 목적함수를 이용해 최적화

Calculate metrics

1. Hits@k: 1 [$RK \leq k$]. Higher is better
2. Reciprocal Rank: $\frac{1}{RK}$. Higher is better

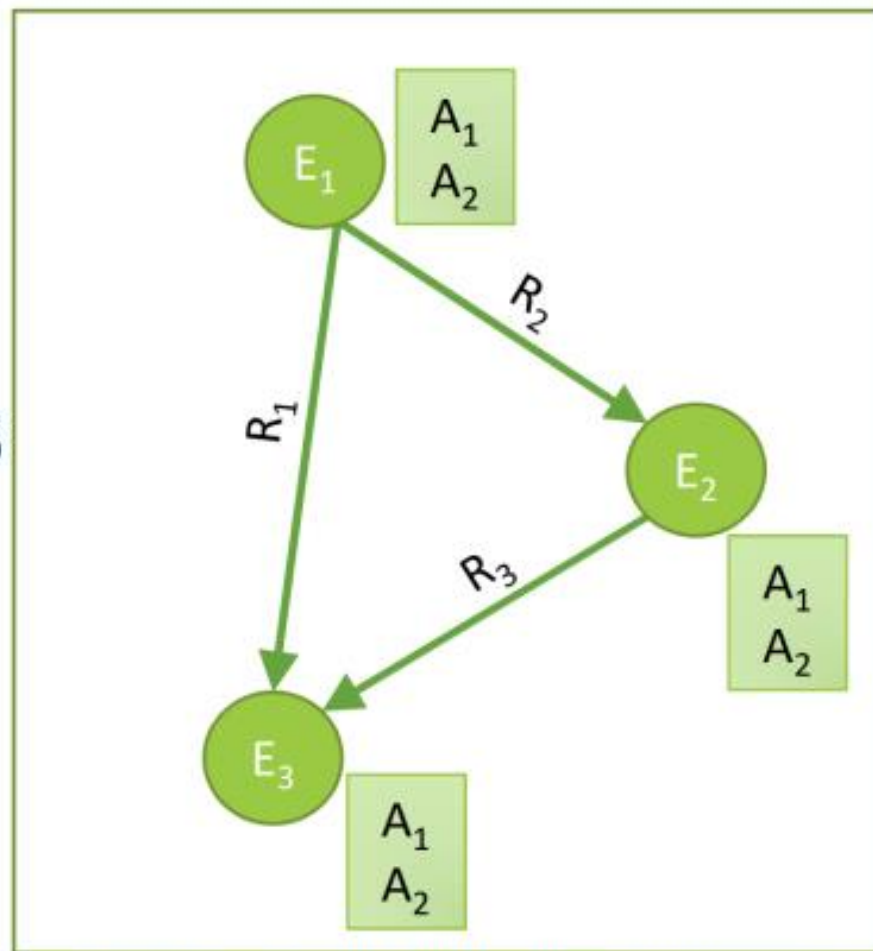
KG



Knowledge Graphs

✓ 지식그래프

- 관련 도메인에서 수집되고 정리된 정보를 heterogeneous graph 형태로 표현했기 때문에 특별히 지식 그래프라고 부른다.



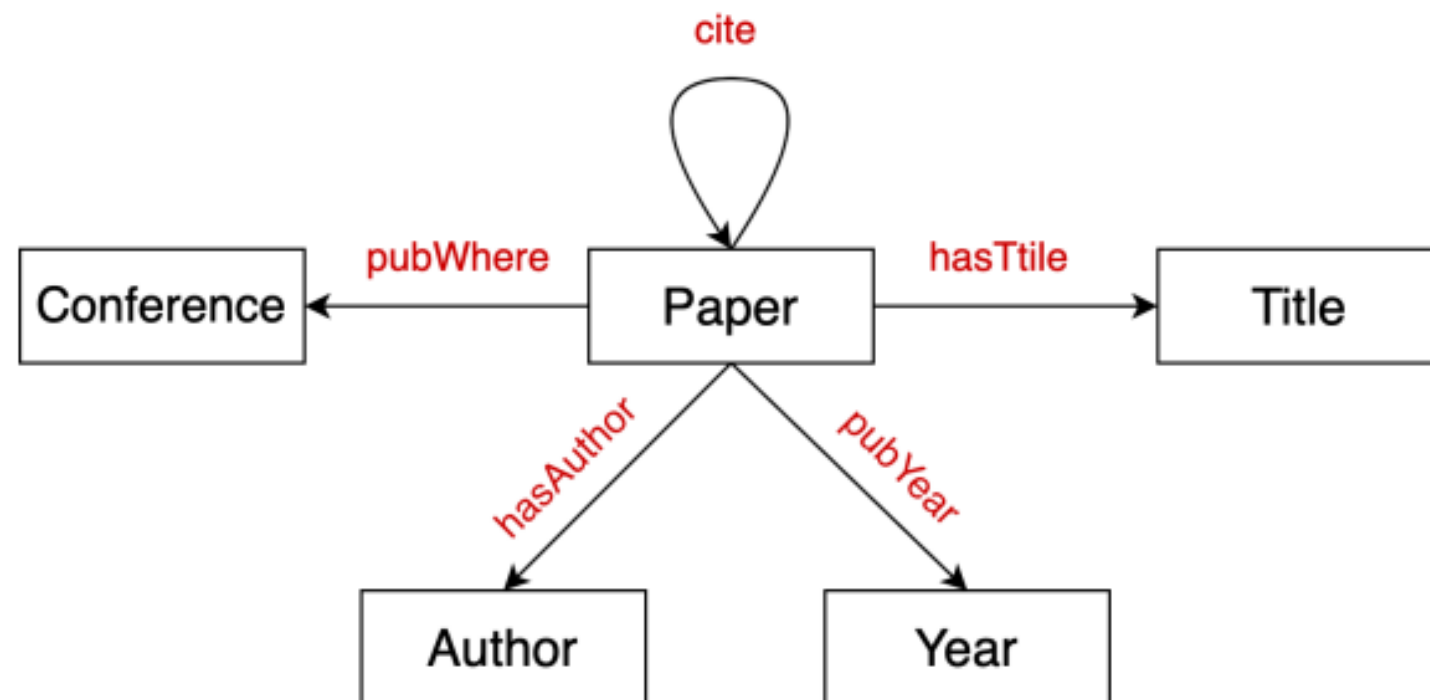
노드들의 종류와 노드 간에 어떠한 관계를 가지는지 다양한 종류의 엣지로 표현이 가능하다.

- Node = entity
- Node는 특정 유형으로 라벨링 된다 (A1,A2)
- Edge 는 두 노드 사이의 관계를 포착한다.
- KG 는 heterogeneous graph 의 예시 중 하나!

Knowledge Graphs

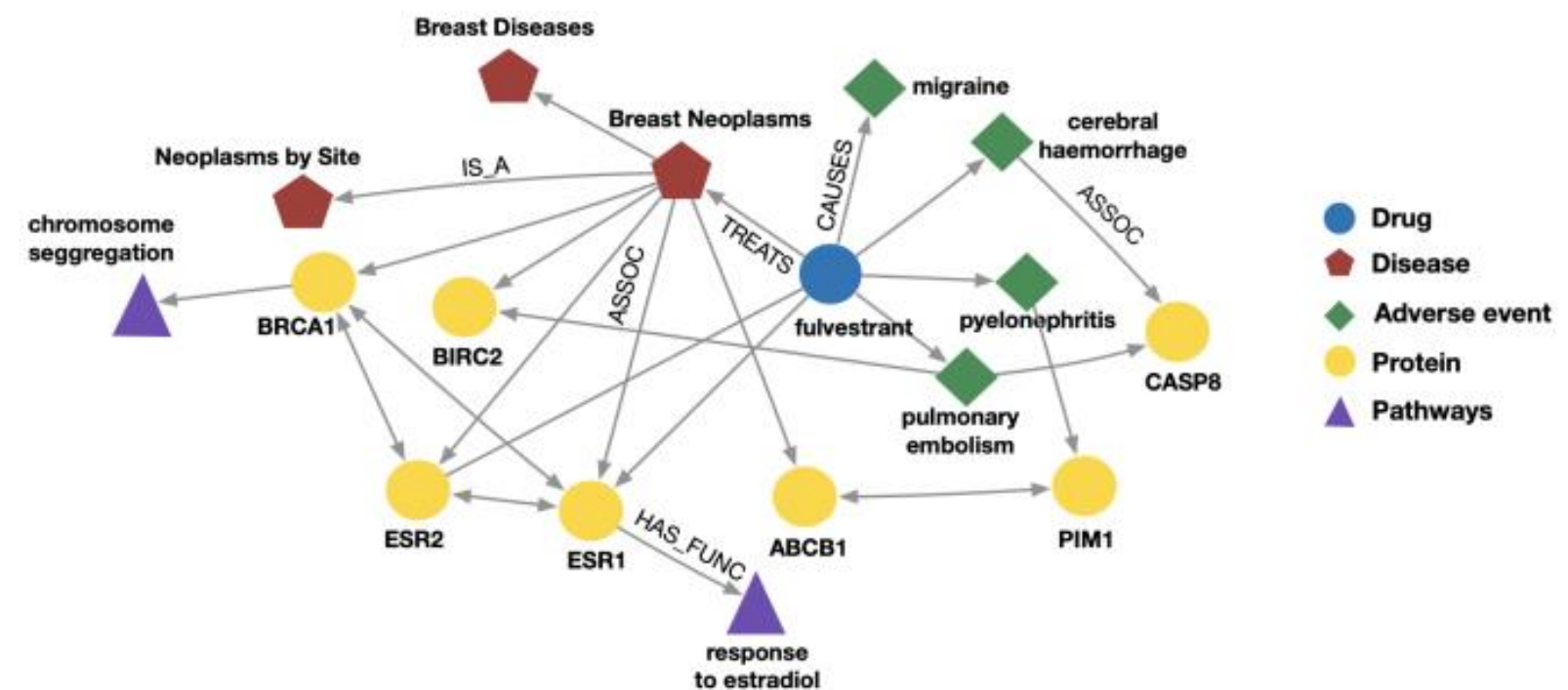
✓ 지식그래프 예시

(1) Bibliographic network



- Node type : 논문, 제목, 저자, 출간일 ...
- Relation type : 출판사, 출간연도, 제목, 인용...

(2) Bio Knowledge Graphs



- Node type : 약, 질병, 단백질경로...
- Relation type : 원인, 연관성, 치료 ...

Knowledge Graphs

✓ 지식그래프 예시

YouTube

사이트



유튜브는 구글이 서비스하는 동영상 공유 플랫폼이다. 2006년에 구글이 인수했고, 전세계 최대 규모의 동영상 공유 및 호스팅 사이트로서, 이용자가 영상을 시청·업로드·공유할 수 있다. 유튜브의 본사는 미국 캘리포니아주 샌브루노에 위치해 있다. [위키백과](#)

수익: 미확인

본사: [미국 캘리포니아 샌브루노](#)

창립: 2005년 2월 14일, [미국 캘리포니아 샌머테이오](#)

창립자: [자베드 카림](#), [채드 헐리](#), [스티브 천](#)

모회사: [구글](#)

CEO: [수전 워치츠키](#) (2014년 2월 5일-)

자회사: [SZS Tech Private Limited](#), [YouTube BrandConnect](#), [더보기](#)

면책조항

• 구글에 특정 키워드를 검색하면, 해당 키워드와 관련된 정보들이 정리된 형태로 표현되는데, 이는 “유튜브” 라는 검색어 노드와 연결된 수많은 노드들을 정리한 것이다 → 지식그래프 형태

- Google Knowledge Graph
- Amazon Product Graph
- Facebook Graph API
- IBM Watson
- Microsoft Satori

• NLP의 QA 가 지식그래프 형태로 정보를 가지고 있음

KG Completion



Knowledge Graphs Completion

✓ 문제제기

- 지식 그래프에 사용되는 정보들은 거대하지만 (Massive), 불완전 (Incomplete) 하다

↳ EX. 대다수의 유저는 자신의 개인정보를 모두 입력하지 않는다.

↳ 출생연도/학교/거주지역 등 중요 노드와 연결되지 않은 유저들이 많음

↳ KG 를 완성도 있게 만들기 위해, 연결 가능성이 높은 엣지를 찾는 것이 중요!

→ KG Completion

Knowledge Graphs Completion

✓ KG Completion

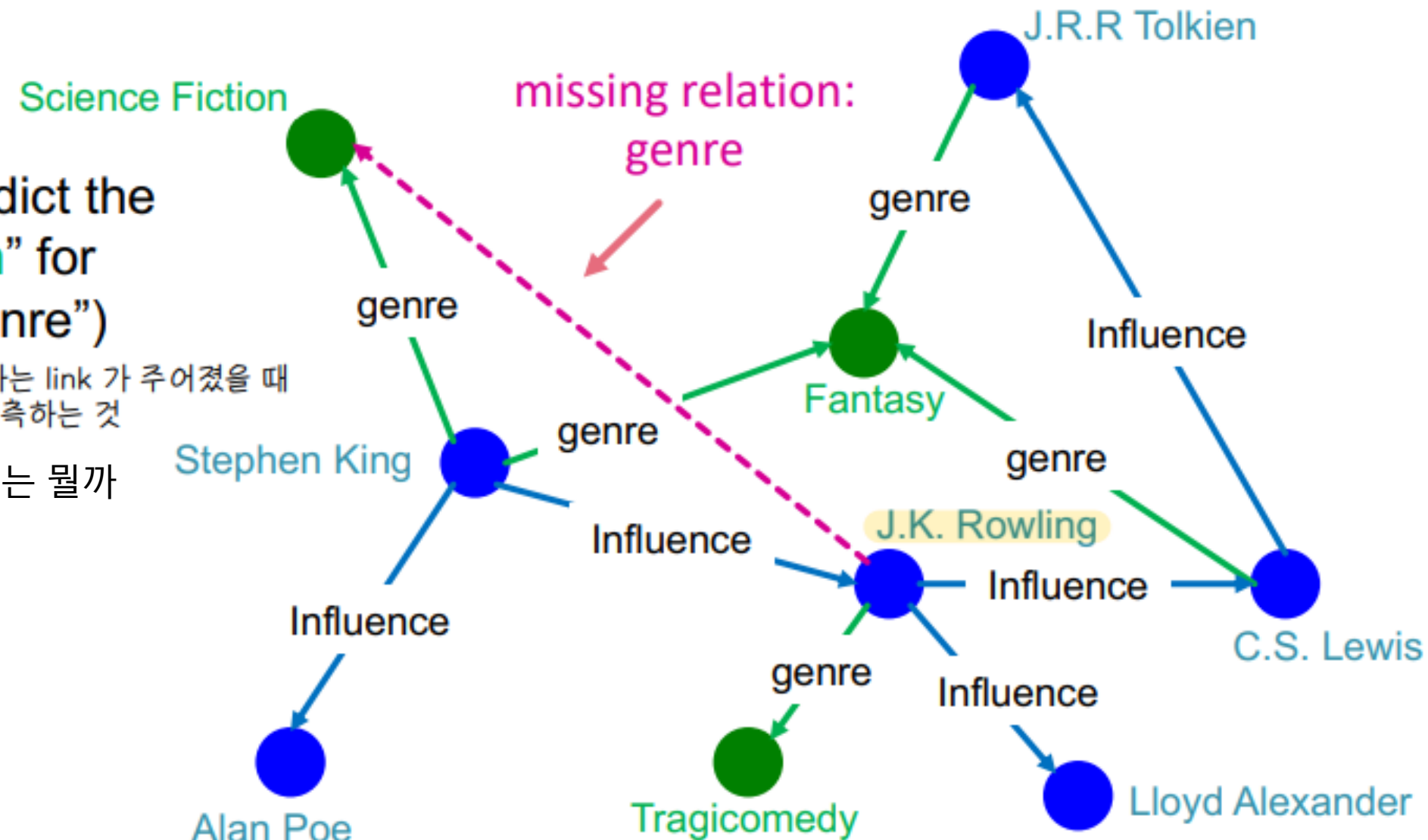
- **노드 하나와 엣지**가 주어진 상태에서 부합하는 노드를 예측

VS link completion: 두 노드가 주어진 상황에서 엣지가 존재할 확률을 계산

Example task: predict the tail “Science Fiction” for (“J.K. Rowling”, “genre”)

jkrowling 이라는 노드와, genre 라는 link 가 주어졌을 때 Science Fiction 이라는 노드를 예측하는 것

JK롤링이 주로 쓰는 장르는 뭘까



Knowledge Graphs Completion

✓ Relation Pattern

KG 에 존재하는 관계들은 저마다 다른 성질을 가진다. 이러한 성질들을 만족하면서, 임베딩 할 수 있다면 최고의 모델이 되는 것임!

① Symmetric

$$r(h, t) \Rightarrow r(t, h)$$

Head 와 tail 이 바뀌어도 동일한 관계가 성립되는 경우
EX. 가족여부, 룸메이트 여부

② Inverse

$$r_2(h, t) \Rightarrow r_1(t, h)$$

Head 와 tail 이 바뀌면 다른 관계가 성립되는 경우
EX. 교수와 학생의 관계

③ Composition

$$r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z)$$

3단 논법으로 설명되는 관계로, 각기 다른 두 관계에 대해 정의된 요소가 3번째 관계로 정의되는 경우
EX. (짱구, 간다) – (슈퍼), (슈퍼, 판다) – 초코비, (짱구, 산다) - 초코비

④ 1-to-N

$$r(h, t_1), r(h, t_2), \dots, r(h, t_n)$$

하나의 head 가 하나의 relation 에 대해 여러 tail 을 가질 때
EX. 지도교수와 지도 석박사 학생들 간의 관계

Knowledge Graphs Completion

✓ KG completion

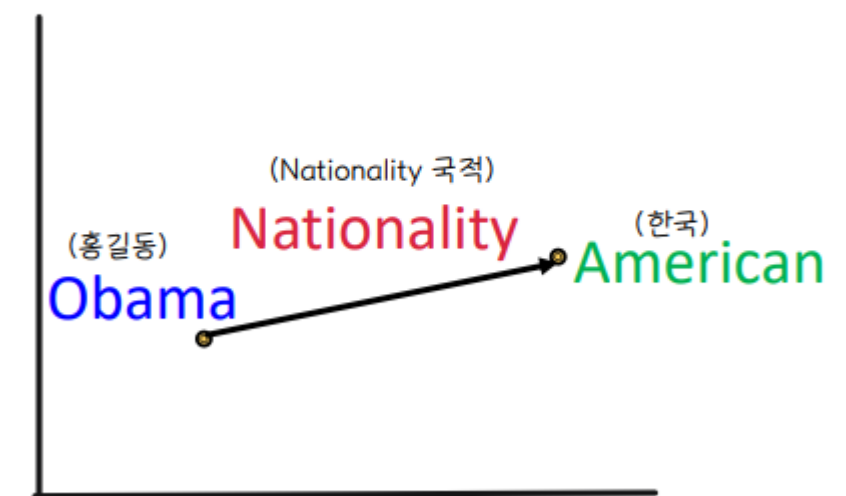
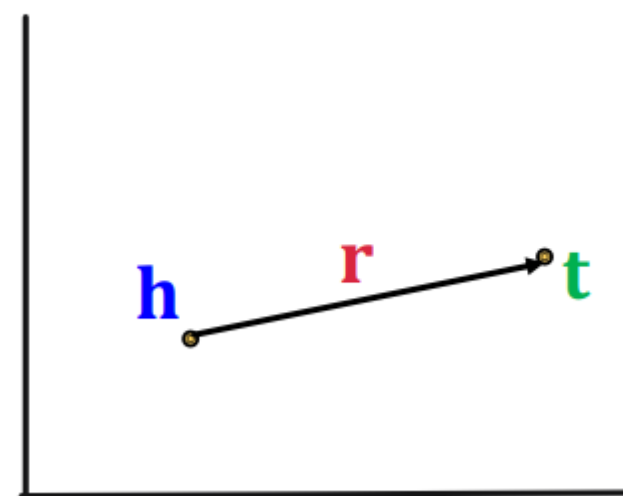
- 표현방식 : $(h, r, t) \rightarrow h$ 노드, r 엣지, t 노드를 어떻게 임베딩 할까 ...
 - ☞ 임베딩 방법 : TransE, TransR, DistMult, ComplEx

① TransE • 임베딩이 옳게 되었다면, 두 단어의 관계는 항상 일정한 벡터로 표현됨

For a triple (h, r, t) , $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$,
 $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ if the given fact is true
else $\mathbf{h} + \mathbf{r} \neq \mathbf{t}$

* $(\mathbf{h} + \mathbf{r} - \mathbf{t}) = 0$ 이 되도록 하는 것을 목표로 한다.

Scoring function: $f_r(h, t) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||$



Knowledge Graphs Completion

② TransR

- TransE 는 모든 노드와 관계를 같은 임베딩 공간에 매핑한다 → symmetric, 1-to-N 성립X

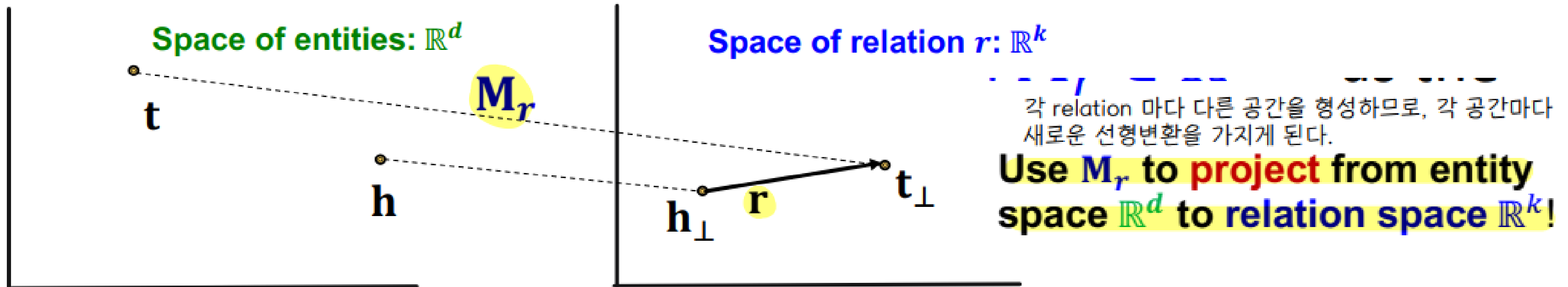
☞ TransR 은 entity 공간과 relation 공간을 분리하여 해결한다.

$$\mathbb{R}^d, \mathbb{R}^k, \mathbf{M}_r \in \mathbb{R}^{k \times d}$$

선형변환

- **Score function:** $f_r(h, t) = -||\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp||$

선형변환으로 두 노드를 relation 공간으로 매핑했을 때,
두 사영이 relation 과 TransE의 관계를 가지도록 한다.



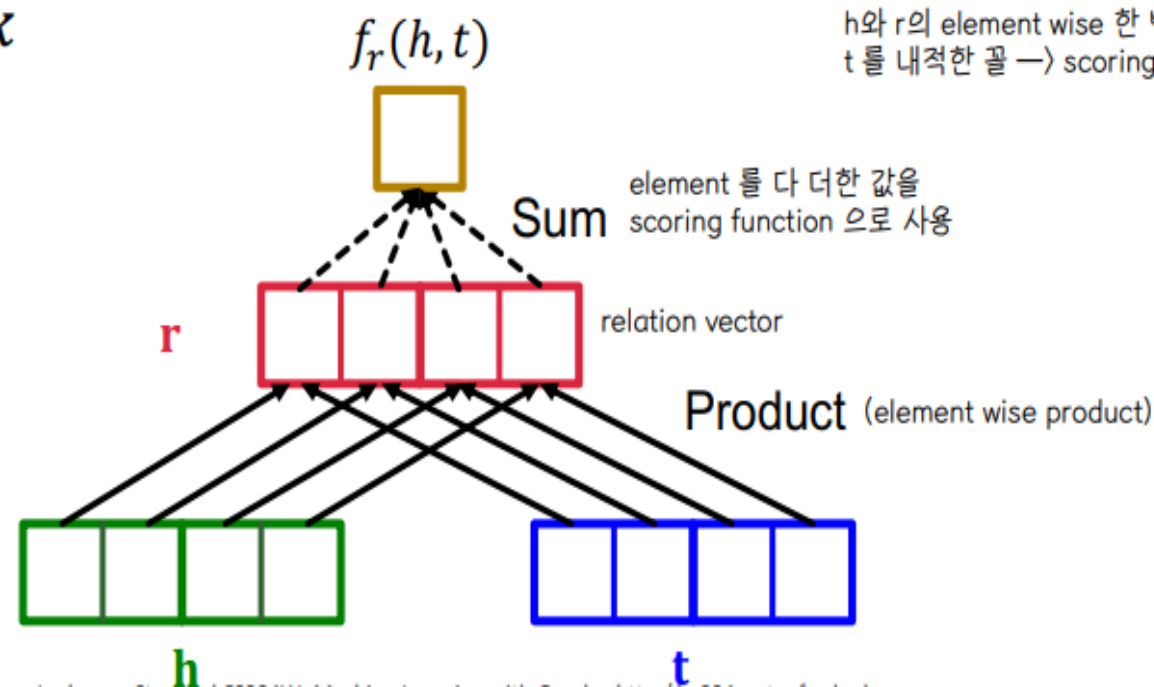
Knowledge Graphs Completion

③ DistMult

- TransE와 TransR 은 모두 벡터 간의 L1, L2 거리를 기반으로 목적함수가 구성됨

↔ DistMult 는 코사인 유사도를 기반으로 목적함수를 구성

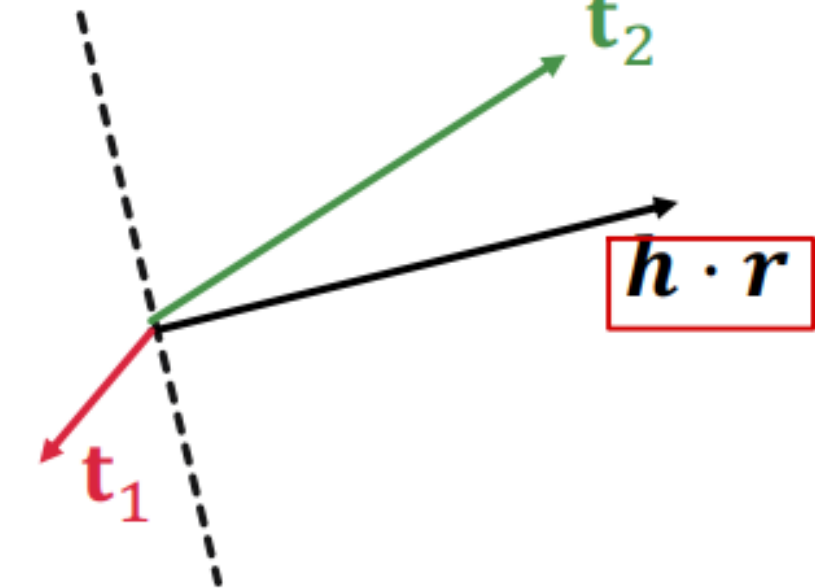
- **Score function:** $f_r(h, t) = \langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle = \sum_i \mathbf{h}_i \cdot \mathbf{r}_i \cdot \mathbf{t}_i$
h와 r의 element wise 한 벡터에 t를 내적한 꼴 → scoring function
- $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$



$$f_r(h, \mathbf{t}_1) < 0,$$

$\mathbf{h} \cdot \mathbf{r}$ 쪽에 \mathbf{t} 가 위치하면 scoring function은 양수가 되고, 반대편에 위치하면 음수가 된다.

$$f_r(h, \mathbf{t}_2) > 0$$



④ ComplEx

- DistMult 와 동일하나, 복소수체에서 임베딩 벡터를 다룸

Knowledge Graphs Completion

Model	Score	Embedding	Sym.	Antisym.	Inv.	Compos.	1-to-N
TransE	$-\ \mathbf{h} + \mathbf{r} - \mathbf{t}\ $	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^k$	✗	✓	✓	✓	✗
TransR	$-\ \mathbf{W}_r \mathbf{h} + \mathbf{r} - \mathbf{W}_r \mathbf{t}\ $	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^k, \mathbf{W}_r \in \mathbb{R}^k$	✓	✓	✓	✗	✓
DistMult	$\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^k$	✓	✗	✗	✗	✓
ComplEx	$\text{Re}(\langle \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} \rangle)$	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{C}^k$	✓	✓	✓	✗	✓

모든 관계를 표현할 수 있는 모델은 없다. 표현할 관계에 따라 모델을 선택해야 한다.
TransE 가 가장 빨라서 기본적으로 시도해보고 다른 것들을 시도해보는 것을 추천!