

# Chapter 5. Knowledge Graph

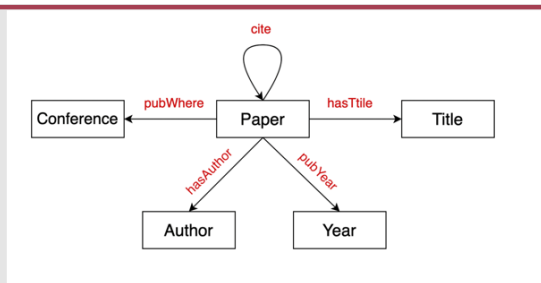
## Outline

- A. Knowledge Graph Embedding
- B. Knowledge Graph Completion
  - a. TransE
  - b. TransR
  - c. DistMult
  - d. ComplEx

## A. Knowledge Graph Embedding

### 1) Definition

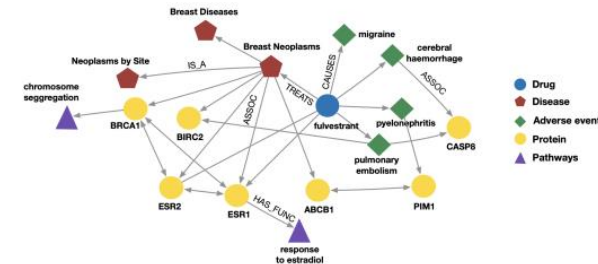
Knowledge Graph는 기존의 GNN 모델로는 다룰 수 없다. 그 이유는 Knowledge Graph는 **Heterogeneous** Graph이기 때문이다. Heterogeneous Graph는 Node(Entity)나 Edge(Relation)의 type이 여러 개인 그래프이다.



- |   |                             |
|---|-----------------------------|
| 1. Node = <b>Entity</b>                   | 1. Node Types               |
| 2. Node → Labeled with their <b>types</b> | – Paper, Title, Author, ... |
| 3. Relation = Domain Information          | 2. Relation Types           |
|   | – pubWhere, pubYear, ...    |

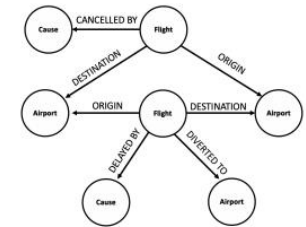
➡ KG = A Kind of H.G

## 2) Examples of KG



### Biomedical Knowledge Graphs

Example node: **Migraine**  
Example edge: (fulvestrant, Treats, Breast Neoplasms)  
Example node type: **Protein**  
Example edge type (relation): **Causes**



### Event Graphs

Example node: **SFO**  
Example edge: (UA689, Origin, LAX)  
Example node type: **Flight**  
Example edge type (relation): **Destination**

## # In Practice

- Google Knowledge Graph
- Amazon Product Graph
- Serving information for a give search keyword
- Question answering and conversation agents

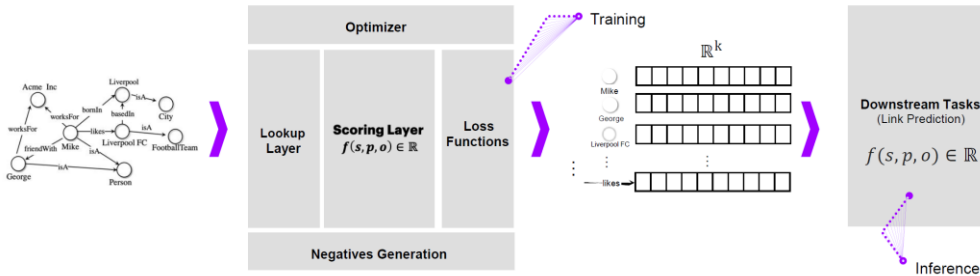
## # Knowledge Graph Benchmarks

FreeBase, WordNet, Wikidata, YAGO, NELL

- a. Massive Graph이다.
- b. Incomplete하다.(many true edges(relations) are missing)
- c. Challenges
  - i. KG is Incomplete → enumerating all the facts is intractable
  - ii. Plausible 하지만 실제로는 유실된 link를 예측할 수 있는가?

### 3) Knowledge Graph Embedding

- Components



Knowledge Graph를 임베딩할 때 엔티티와 릴레이션들을 연속된 저차원의 벡터 공간으로 projection하는 것이다. Knowledge Graph Embedding은 Supervised learning이며 Automatic하다.

#### # Model

- TransE / b. TransR / c. DistMult / d. ComplEx

이 4가지 모델들은 서로 다른 geometric intuitions을 기반으로 한 모델들이다. Relation들의 type을 포착하며 이를 통해 Link prediction을 진행한다.

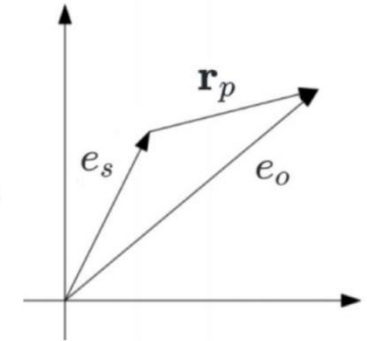
- Input: Knowledge Graph (KG) G
- Scoring function for a triple  $f(t)$
- Loss function L
- Optimization algorithm
- Negative generation strategy

#### # Scoring function

- Translation-Based scoring functions

##### a. TransE

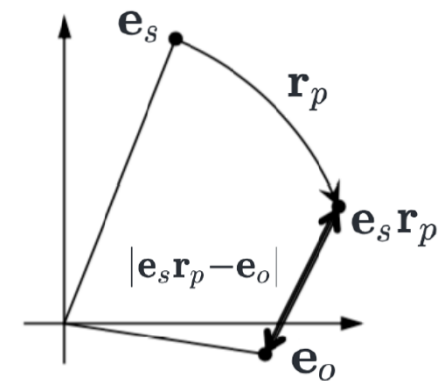
$$f_{TransE} = -||(\mathbf{e}_s + \mathbf{r}_p) - \mathbf{e}_o||_n$$



##### b. RotatE

릴레이션들이 complex space에 모델링된다. Complex embedding 간에 Element-Wise product 연산을 한다.

$$f_{RotatE} = -||\mathbf{e}_s \circ \mathbf{r}_p - \mathbf{e}_o||_n$$



## # Loss function

- Pairwise margin-based hinge loss

Pays a penalty if score of positive triple < score of synthetic negative by a margin  $\gamma$

$$\mathcal{L}(\Theta) = \sum_{t^+ \in \mathcal{G}} \sum_{t^- \in \mathcal{C}} \max(0, [\gamma + \underbrace{f(t^-; \Theta)}_{\substack{\text{Score} \\ \text{assigned to a} \\ \text{synthetic} \\ \text{negative}}} - \underbrace{f(t^+; \Theta)}_{\substack{\text{Score} \\ \text{assigned to a} \\ \text{true triple}}}]])$$

- Negative log-likelihood / Cross entropy

$$\mathcal{L}(\Theta) = \sum_{t \in \mathcal{GUC}} \log(1 + \exp(-y f(t; \Theta)))$$

Label of the triple  $t$       $y \in \{-1, 1\}$

- Binary cross entropy

$$\mathcal{L} = -\frac{1}{N} \sum_{t \in \mathcal{GUC}}^N y \cdot \log(\sigma(f(t; \Theta))) + (1 - y) \cdot \log(1 - \sigma(f(t; \Theta)))$$

- Self-Adversarial

$$\mathcal{L} = -\log \sigma(\gamma + f(t^+; \Theta)) - \sum_{t \in \mathcal{G}}^N \underbrace{p(t^-; \Theta)}_{\substack{\text{Weight for the negative sample } t}} \log \sigma(-f(t^-; \Theta) - \gamma)$$

## # Regularization

- L1, L2, L3 normalization
- Dropout(ConvE)

## # Initialization

- Random(uniform)
- Random(normal)
- Glorot
- (Glorot | Xavier initialization)

### Forward Pass Equation

$$\text{Var}[z_k^{i+1}] = \text{Var}[f(z^i W_{:,k}^i + b_k^i)] \Rightarrow \text{Var}[W^i] = \frac{1}{n^i}$$

### Backward Pass Equation

$$\text{Var}\left[\frac{\partial \mathcal{L}}{\partial s_k^i}\right] = \text{Var}\left[\sum_{j=1}^{n^{i+2}} W_{j,k}^{i+1} \frac{\partial \mathcal{L}}{\partial s_k^{i+1}} f'(s_k^i)\right] \Rightarrow \text{Var}[W^i] = \frac{1}{n^{i+1}}$$

### Weight Distributions

$$\text{Var}[W^i] = \frac{2}{n^i + n^{i+1}} \Rightarrow \begin{cases} W^i \sim \mathcal{N}(0, \sigma^2) \Rightarrow \sigma = \sqrt{\frac{2}{n^i + n^{i+1}}} \\ W^i \sim \mathcal{U}(-a, a) \Rightarrow a = \sqrt{\frac{6}{n^i + n^{i+1}}} \end{cases}$$

Xavier Initialization -> [\[Link\]](#)

## # Negative Example Generation

False Positive로부터 Negative가 만들어진다.

$$\mathcal{C} = \{(\hat{s}, p, o) | \hat{s} \in \mathcal{E}\} \cup \{(s, p, \hat{o}) | \hat{o} \in \mathcal{E}\}$$

The predicate is unaltered

"corrupted subject"      "corrupted" object

Knowledge Graph는 거시적으로 봤을 때 incomplete하지만, 미시적으로 locality에서 보자면 complete하다. 이로 인해, synthetic negative triple이 만들어질 수 있다.

## # Synthetic negative triple

- $\mathcal{E} = \{Mike, Liverpool, AcmeInc, George, LiverpoolFC\}$
- $\mathcal{R} = \{bornIn, friendWith\}$
- $t \in G = (Mike \quad bornIn \quad Liverpool)$
- $\mathcal{C}_t =$ 

(Mike	bornIn	AcmeInc)
(Mike	bornIn	LiverpoolFC)
(George	bornIn	Liverpool)
(AcmeInc	bornin	Liverpool)

## # Training Procedure and Optimizer

- Optimizer:** learn optimal parameters (e.g., embeddings). Off-the-shelf SGD variants (AdaGrad, Adam)

$$\min_{\theta} \mathcal{L}(\theta)$$

- Reciprocal triples:**  
injection of reciprocal triples in training set

`<Alice childOf Jack>` [Dettmers et al. 2017]  
`<Jack childOf† Alice>` [Lacroix et al. 2018]

## # Evaluation Metric

MR, MRR, Hits@N

- Mean Rank (MR)

$$MR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} rank_{(s,p,o)_i}$$

- Mean Reciprocal Rank (MRR)

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_{(s,p,o)_i}}$$

- Hits@N

$$Hits@N = \frac{1}{|Q|} \sum_{i=1}^{|Q|} 1 \text{ if } rank_{(s,p,o)_i} \leq N$$

## Example

s	p	o	score	rank
Mike	born_in	Leeds	0.789	1
Mike	born_in	Liverpool	0.753	2
Mike	born_in	Germany	0.695	3
George	born_in	Liverpool	0.456	4
Mike	born_in	George	0.234	5

- How unseen, test positive triples rank against synthetic negatives?

s	p	o	score	rank
Mike	friend_with	George	0.901	1
Mike	friend_with	Jim	0.345	2
Acme	friend_with	George	0.293	3
Mike	friend_with	Liverpool	0.201	4
France	friend_with	George	0.156	5

Unseen positive triples (test set)

$MR = 1.5$   
 $MRR = 0.75$   
 $Hits@1 = 0.5$   
 $Hits@3 = 1.0$

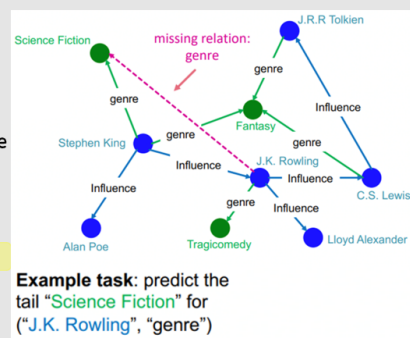
## B. Knowledge Graph Completion

### 1) Definition

- Slightly different from Link prediction

- ✓ **Link Prediction**
- ✓ Nothing is given, just predict what links are missing in the entire graph

- ✓ **Knowledge Graph Completion**
- ✓ For a given (head, relation), predict missing tails



Knowledge Graph Completion은 Link prediction과 유사하다. 하지만, 아무것도 주어지지 않는 조건에서 링크만을 찾는 Link prediction과는 다르게 Knowledge Graph Completion (KGC)에서는 **Given (head, relation)에 대해 tail 엔티티**를 찾는 task이다. (Triple = (head, relation, tail))

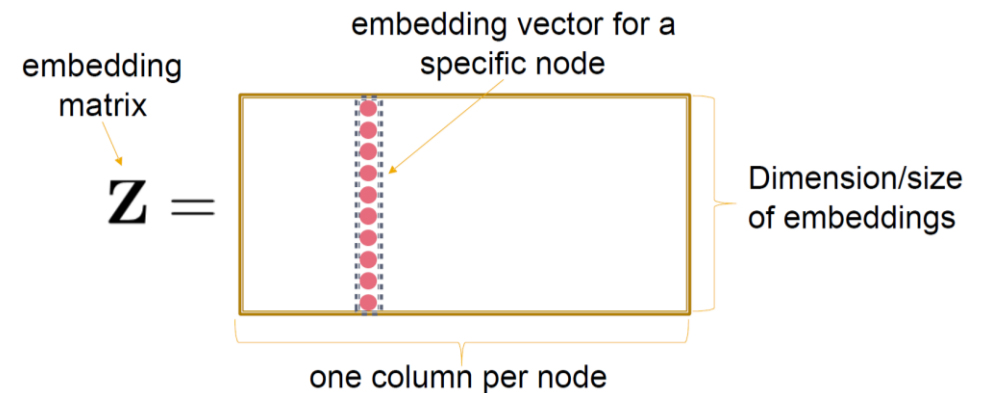
Given: Head(Subject), relation(predicate), tail(Object)

$\langle h, r, t \rangle$  or  $\langle s, p, o \rangle$

→ Find the missing tail  $t$

### 2) Shallow Embedding

KGC의 가장 간단한 임베딩 방식은 Shallow Embedding이다. 인코더가 가장 간단하게 임베딩을 찾아보는 방식이다. (Encoder is just an embedding lookup)



Knowledge Graph Completion에서는 head 엔티티와 relation이 주어졌을 때 tail을 찾는 것이다. 따라서 모델들이 1)어떻게  $(h, r)$ 를 임베딩하고, 2)어떻게 closeness를 정의하는가에 주목해야 한다.

### Knowledge Graph Representation

- ✓ Triples  $(h, r, t)$
- ✓  $h = \text{head}, r = \text{relation}, t = \text{tail}$

### Key Idea of KG Completion

- ✓ Model entities and relations in the embedding/vector space  $\mathbb{R}^d$

Entities & Relations → Shallow Embedding

- ✓ Given a true triple  $(h, r, t)$
- ✓ the embedding of  $(h, r)$  should be close to the embedding of  $t$

- ✓ How to embed  $(h, r)$ ?
- ✓ How to define closeness?

### Connectivity Patterns

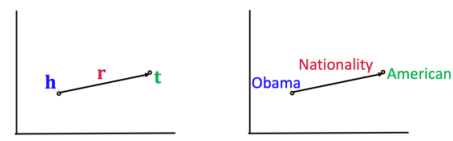
Symmetric relation	$[\langle h, r \rangle = t] = [\langle t, r \rangle = h]$
Antisymmetric relation	$[\langle h, r \rangle = t] \neq [\langle t, r \rangle = h]$
Inverse relation	$\langle h, r_1, t \rangle = \langle t, r_2, h \rangle$
Composition relation	$r_1(x, y) \wedge r_2(y, z) = r_3(x, z)$
1-to-N-relation	$r(h, t_1), r(h, t_2), \dots, r(h, t_n)$

## C. Knowledge Graph Models

### 1) TransE

Scoring function:  $f_r(h, t) = -||h + r - t||$

Scoring function:  $f_r(h, t) = -||h + r - t||$



<minimizing Loss>

관계가 성립하는  
t에 대해서 거리 최소화(min d)

Goal: triple(h, r, t),  $h + r \approx t$

Contrastive loss

$$\sum_{((h, l, t), (h', l, t')) \in T_{batch}} \nabla[\gamma + d(h + l, t) - d(h' + l, t')]_+$$

Positive Sample

Negative Sample

d: Distance

관계가 성립하지 않는  
t'에 대해서 거리 최대화(Max d)

Scoring function은 쉽게 말하면 Loss 식에 들어가는 term이다. TransE 모델은 Knowledge Graph Completion을 하는 모델 중 가장 간단한 모델이다. head 엔티티와 relation, tail 엔티티의 관계를 단순히 하나의 vector space에서 정의했다. 이 때 이러한 관계를 정의한 것을 Translation function이라고 하고, TransE, TransR 같은 모델을 Translation method model이라고 한다.

#### (1) Translation function of TransE

$$h + r \approx t \text{ (if the given fact is true)}$$

#### (2) Scoring function of TransE

$$f_r(h, t) = -||h + r - t||$$

#### Algorithm 1 Learning TransE

input Training set  $S = \{(h, \ell, t)\}$ , entities and rel. sets  $E$  and  $L$ , margin  $\gamma$ , embeddings dim.  $k$ .

```

1: initialize  $\ell \leftarrow \text{uniform}(-\frac{\gamma}{\sqrt{k}}, \frac{\gamma}{\sqrt{k}})$  for each  $\ell \in L$ 
2:  $\ell \leftarrow \ell / ||\ell||$  for each  $\ell \in L$ 
3:  $e \leftarrow \text{uniform}(-\frac{\gamma}{\sqrt{k}}, \frac{\gamma}{\sqrt{k}})$  for each entity  $e \in E$ 
4: loop
5:    $e \leftarrow e / ||e||$  for each entity  $e \in E$ 
6:    $S_{batch} \leftarrow \text{sample}(S, b)$  // sample a minibatch of size  $b$ 
7:    $T_{batch} \leftarrow \emptyset$  // initialize the set of pairs of triplets
8:   for  $(h, \ell, t) \in S_{batch}$  do
9:      $(h', \ell', t') \leftarrow \text{sample}(S'_{(h, \ell, t)})$  // sample a corrupted triplet
10:     $T_{batch} \leftarrow T_{batch} \cup \{(h, \ell, t), (h', \ell', t')\}$ 
11:   end for
12:   Update embeddings w.r.t.  $\sum_{((h, \ell, t), (h', \ell', t')) \in T_{batch}} \nabla[\gamma + d(h + \ell, t) - d(h' + \ell', t')]_+$ 
13: end loop
```

Entities and relations are initialized uniformly, and normalized

Negative sampling with triplet that does not appear in the KG

$d$  represents distance (negative of score)

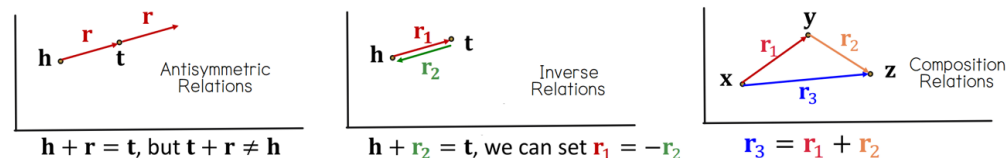
Contrastive loss: favors lower distance (or higher score) for valid triplets, high distance (or lower score) for corrupted ones

### (3) Relation Patterns

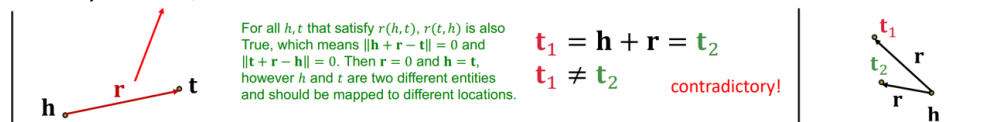
#### Connectivity Patterns

Symmetric relation	$[\langle h, r \rangle = t] = [\langle t, r \rangle = h]$
Antisymmetric relation	$[\langle h, r \rangle = t] \neq [\langle t, r \rangle = h]$
Inverse relation	$\langle h, r_1, t \rangle = \langle t, r_2, h \rangle$
Composition relation	$r_1(x, y) \wedge r_2(y, z) = r_3(x, z)$
1-to-N-relation	$r(h, t_1), r(h, t_2), \dots, r(h, t_n)$

Antisymmetric, Inverse, Composition은 TransE와 적합



Symmetric, 1-to-N 부적합  
only if  $r = 0$ ,  $h = t$





## 2) TransR

TransR은 엔티티와 릴레이션을 같은 벡터 공간에 두는 TransE의 단순성을 지적하며 나온 모델이다. 너무 단순하기 때문에 좋은 성능을 내지 못한다는 것이다. 따라서, **엔티티와 릴레이션을의 공간을 분리하는 것이다.**

1. Dividing Space

- Entity Space  $h, t \in \mathbb{R}^d$
- Relation Space  $r \in \mathbb{R}^k$

2. Goal

Entity  $\xrightarrow[\text{Mapping}]{\text{Linear Transform}}$  Relation Space  
 $(M_r \in \mathbb{R}^{k \times d})$

- Each relation has different relation space  $\rightarrow M_r$
- Score function  $f_r(h, t) = -||h_{\perp} + r - t_{\perp}||$ , Projected entity:  $h_{\perp}$  &  $t_{\perp}$

## 3) DistMult

DistMult는 Bi-linear 방식을 채택한 모델이다. 간단하게 말하면 Scoring function을 정의할 때 head와 relation, tail 임베딩의 값을 모두 Element-wise product한 후 summation한 것이다. 일종의 벡터 3개를 내적한 것과 같다.

\* Scoring function

- cosine similarity between  $h \cdot r$  and  $t$
- $f_r(h, t) = \langle h, r, t \rangle = \sum_i h_i \cdot r_i \cdot t_i$
- $h, r, t \in \mathbb{R}^k$

Element Wise Product  $h \cdot r$  & Tail Entity  $t$

$\xrightarrow[\text{cos similarity}]{\text{Inner Product}}$

Bilinear Modeling

$f_r(h, t_1) < 0, f_r(h, t_2) > 0$

Symmetric Relations

$r = 0, h_{\perp} = M_r h = M_r t = t_{\perp}$

1-to-N Relations

can learn  $M_r$  so that  $t_{\perp} = M_r t_1 = M_r t_2$

Antisymmetric Relations

$r \neq 0, M_r h + r = M_r t$ , Then  $M_r t + r \neq M_r h$

Inverse Relations

$r_2(h, t) \Rightarrow r_1(t, h), r_2 = -r_1, M_{r_1} = M_{r_2}$   
Then  $M_{r_1} t + r_1 = M_{r_1} h$  and  $M_{r_2} h + r_2 = M_{r_2} t$

1-to-N relations

$\langle h, r, t_1 \rangle = \langle h, r, t_2 \rangle$

Symmetric relations

$f_r(h, t) = \langle h, r, t \rangle = \sum_i h_i \cdot r_i \cdot t_i = \langle t, r, h \rangle = f_r(t, h)$

- 1-to-N relations.
- Symmetric 표현 적합

- Antisymmetric.
- Composition.
- Inverse 표현 부적합

Antisymmetric

$f_r(h, t) = \langle h, r, t \rangle = \langle t, r, h \rangle = f_r(t, h) \times$

- $r(h, t)$  and  $r(t, h)$  always have same score!

Inverse

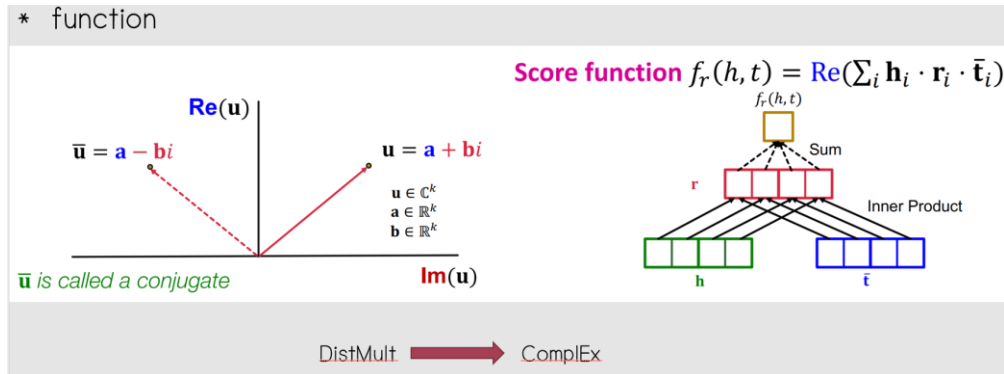
$f_{r_2}(h, t) = \langle h, r_2, t \rangle = \langle t, r_1, h \rangle = f_{r_1}(t, h)$

- This means  $r_2 = r_1$
- But semantically this does not make sense: The embedding of "Advisor" should not be the same with "Advisee".

Composition

#### 4) ComplEx

DistMult를 복소 평면으로 확장한 방식이다. 마찬가지로 Bi-linear 모델을 채택하며, 다만 임베딩 벡터들이 Complex number이기 때문에 real part만 고려해서 scoring function을 정의한다.



#### • Connectivity Patterns

적합

- 1-to-N(Like DistMult)
- Antisymmetric(Conjugate Transpose)
- Symmetric(if imaginary part = 0)
- Inverse

부적합

- Composition(Like DistMult)

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	$\text{Re}(\langle \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} \rangle)$	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{C}^k$	✓	✓	✓	×	✓