Chapter 5. Knowledge Graph

Outline

A. Knowledge Graph Embedding

B. Knowledge Graph Completion

  a. TransE

  b. TransR

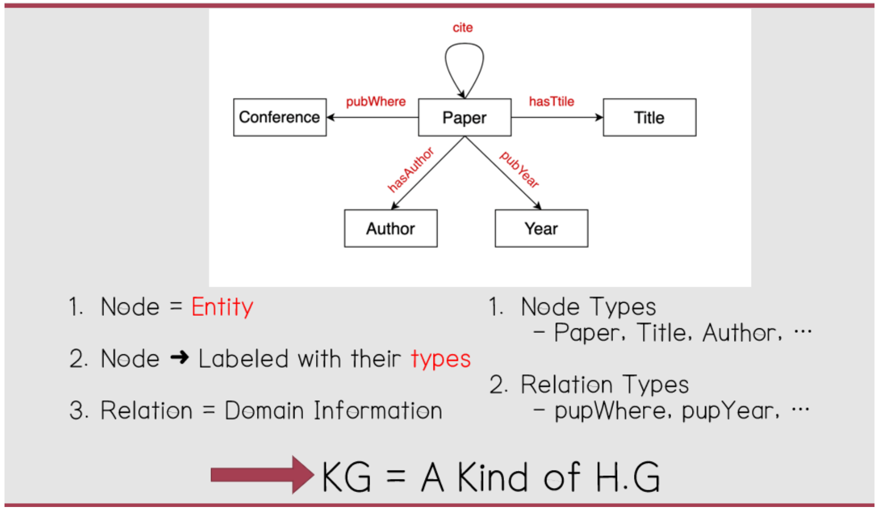
  c. DistMult

  d. ComplEx

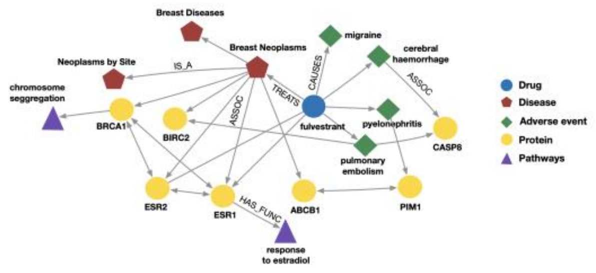
A. Knowledge Graph Embedding

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.

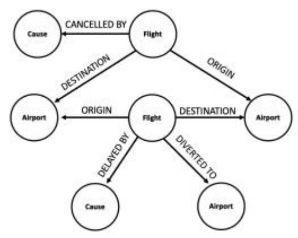


2) Examples of KG



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)

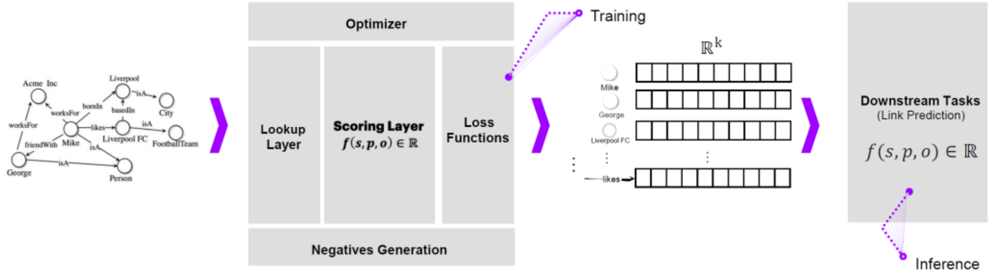
  c. Challenges

    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

a. 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.

a. Input: Knowledge Graph (KG) G

b. Scoring function for a triple f(t)

c. Loss function L

d. Optimization algorithm

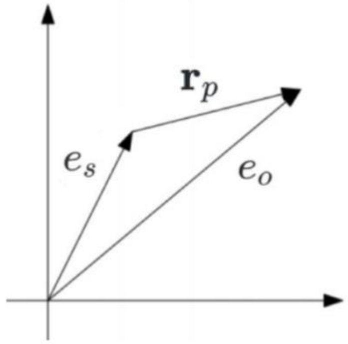
e. Negative generation strategy

Scoring function

- Translation-Based scoring functions

a. TransE

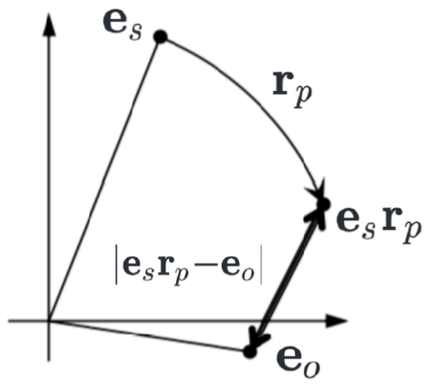
$$f_{TransE} = -||(\mathbf{e}_s + \mathbf{r}_p) - \mathbf{e}_o||_n$$



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

$$\mathcal{L}(\Theta) = \sum_{t^+ \in \mathcal{G}} \sum_{t^- \in \mathcal{C}} \overbrace{max(0, [\gamma + \underbrace{f(t^-;\Theta)}_{\substack{\text{Score}\\ \text{assigned to a}\\ \textbf{synthetic}\\ \textbf{negative}}} - \underbrace{f(t^+;\Theta)}_{\substack{\text{Score}\\ \text{assigned to a}\\ \textbf{true}\text{ triple}}}])}^{\text{Pays a penalty if score of positive triple < score of synthetic negative by a margin } \gamma}$$

- Negative log-likelihood / Cross entropy

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

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

- Binary cross entropy

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

- Self-Adversarial

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

Regularization

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

Initialization

- Random(uniform)
- Random(normal)
- Glorot

  (Glorot이 Xavier intialization이다.)

Forward Pass Equation

$$Var[z_k^{i+1}] = Var[f(z^i W^i_{,jk} + b^i_k)] \Rightarrow Var[W^i] = \frac{1}{n^i}$$

Backward Pass Equation

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

Weight Distributions

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

Xavier Initialization -> [Link]

Negative Example Generation

False Positive로부터 Negative가 만들어진다.

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

"corrupted subject"

"corrupted" object

The predicate is unaltered

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 bornIn 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^{-1} Alice> [Lacroix et al. 2018]

Evaluation Metric

MR, MRR, Hits@N

- Mean Rank (MR)

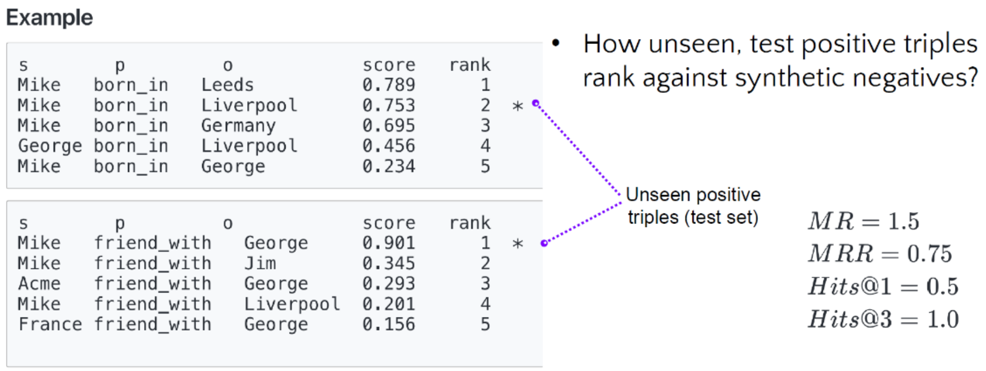
$$MR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} rank_{(s,p,o)_i}$$

- Mean Reciprocal Rank (MRR)

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

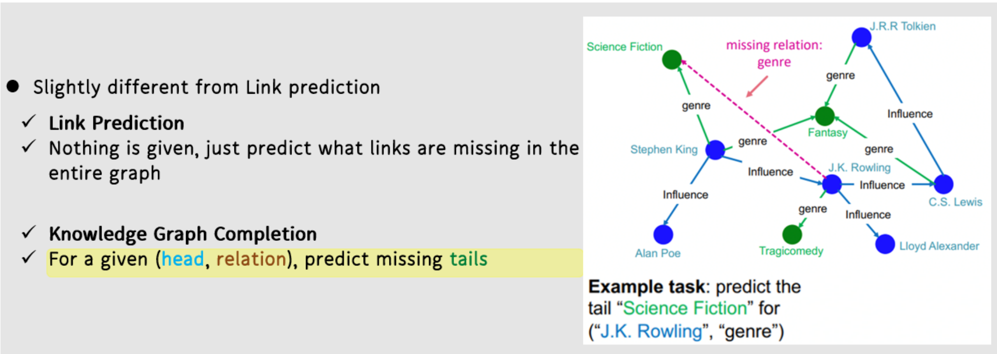
- Hits@N

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



B. Knowledge Graph Completion

1) Definition



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 엔티티를 찾는 task이다. (Triple = **(head, relation, tail)**)

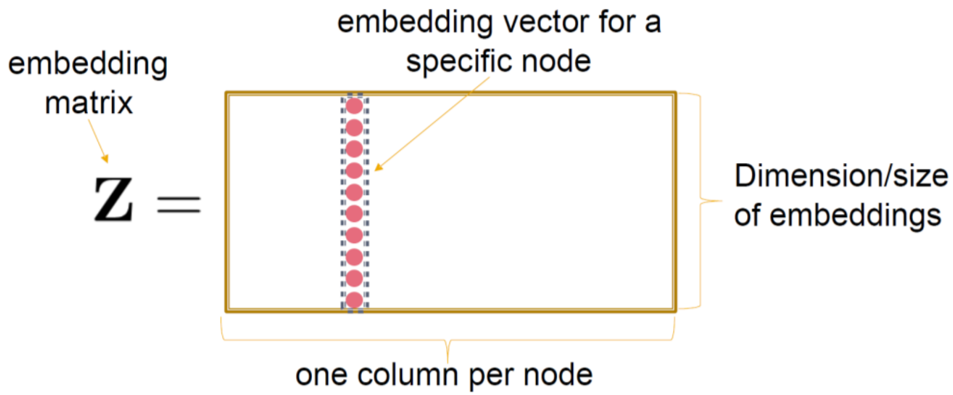
Given: Head(Subject), relation(predicate), tail(Object)

<h,r,t> or <s,p,o>

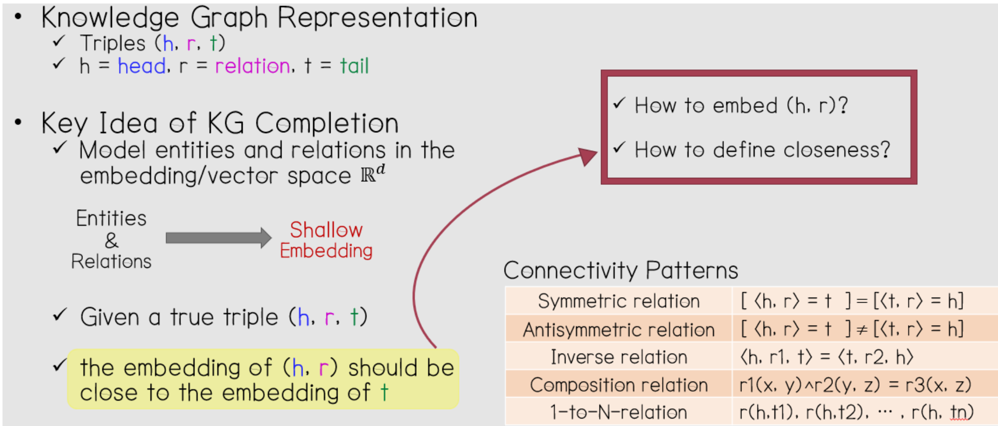
-> Fine the missing tail t

2) Shallow Embedding

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

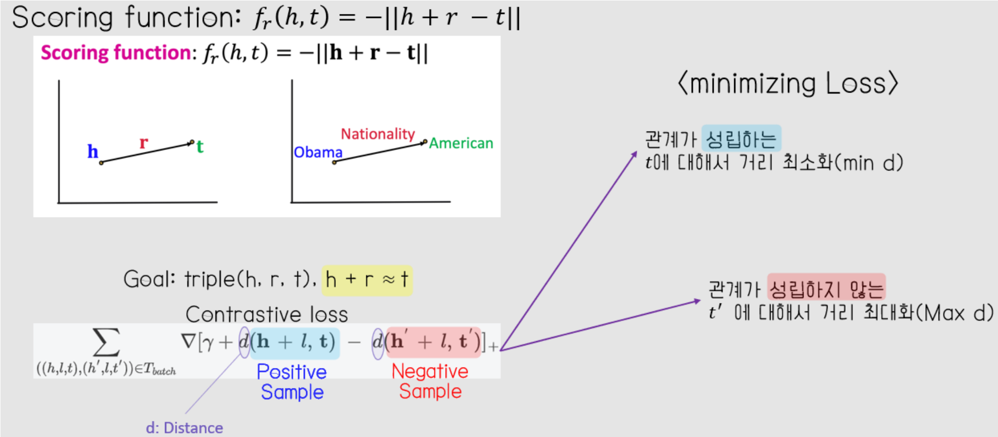


Knowledge Graph Completion에서는 head 엔티티와 relation이 주어졌을 때 tail을 찾는 것이다. 따라서 모델들이 1)어떻게 (h,r)을 임베딩하고, 2)어떻게 closeness를 정의하는가에 주목해야 한다.



C. Knowledge Graph Models

1) TransE



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 단순하게 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 ≈ t (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 γ, embeddings dim. k.

```
 initialize ℓ ← uniform(−6/√k, 6/√k) for each ℓ ∈ L
            ℓ ← ℓ/ ||ℓ|| for each ℓ ∈ L
            e ← uniform(−6/√k, 6/√k) for each entity e ∈ E
 loop
   e ← e/ ||e|| for each entity e ∈ E
   S_batch ← sample(S, b) // sample a minibatch of size b
   T_batch ← ∅ // initialize the set of pairs of triplets
   for (h, ℓ, t) ∈ S_batch do
     (h′, ℓ, t′) ← sample(S′_(h,ℓ,t)) // sample a corrupted triplet
     T_batch ← T_batch ∪ {((h, ℓ, t), (h′, ℓ, t′))}
   end for
   Update embeddings w.r.t.  Σ_((h,ℓ,t),(h′,ℓ,t′))∈T_batch ∇[γ + d(h + ℓ, t) − d(h′ + ℓ, t′)]_+
 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)

Positive sample

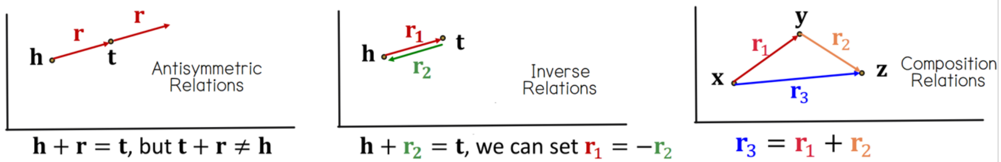
Negative sample

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

(3) Relation Patterns

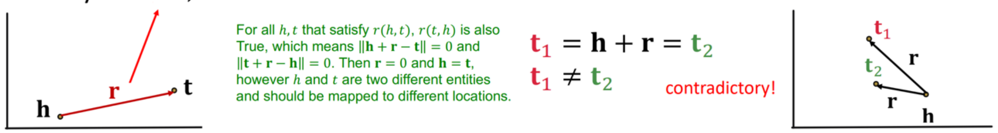
Connectivity Patterns	
Symmetric relation	[⟨h, r⟩ = t] = [⟨t, r⟩ = h]
Antisymmetric relation	[⟨h, r⟩ = t] ≠ [⟨t, r⟩ = h]
Inverse relation	⟨h, r1, t⟩ = ⟨t, r2, h⟩
Composition relation	r1(x, y)∧r2(y, z) = r3(x, z)
1-to-N-relation	r(h,t1), r(h,t2), ⋯ , r(h, tn)

- Antisymmetric, Inverse, Composition은 TransE와 적합



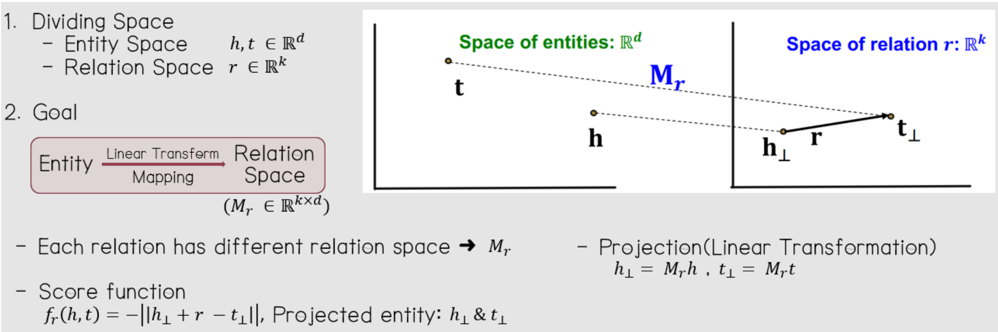
- Symmetric, 1-to-N 부적합

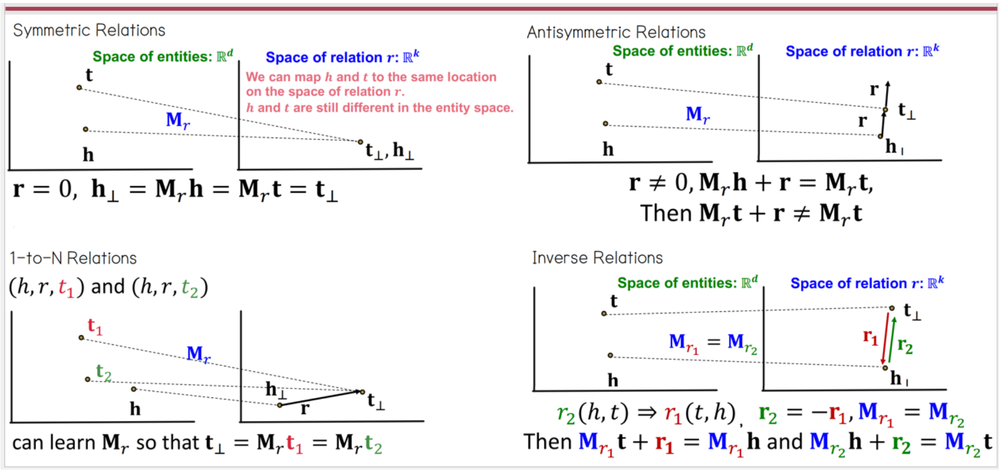
only if $\mathbf{r} = 0$, $\mathbf{h} = \mathbf{t}$



2) TransR

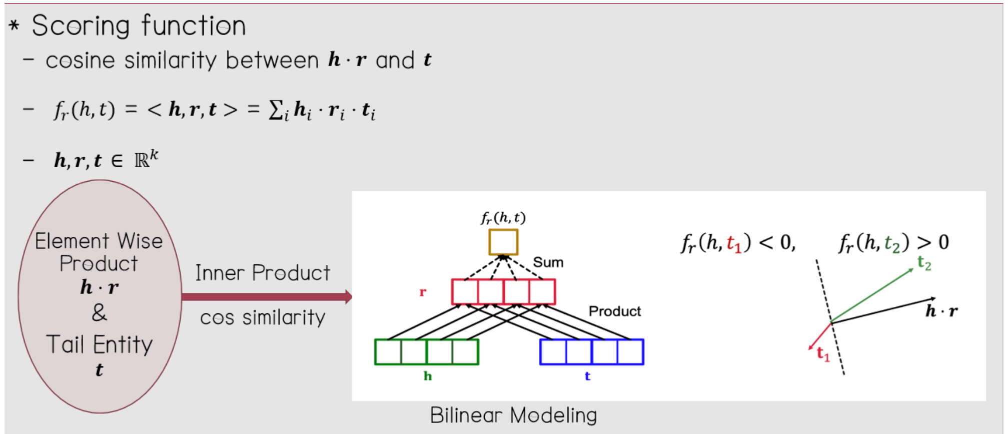
TransR은 엔티티와 릴레이션을 같은 벡터 공간에 두는 TransE의 단순성을 지적하며 나온 모델이다. 너무 단순하기 때문에 좋은 성능을 내지 못한다는 것이다. 따라서, 엔티티와 릴레이션의 공간을 분리하는 것이다.

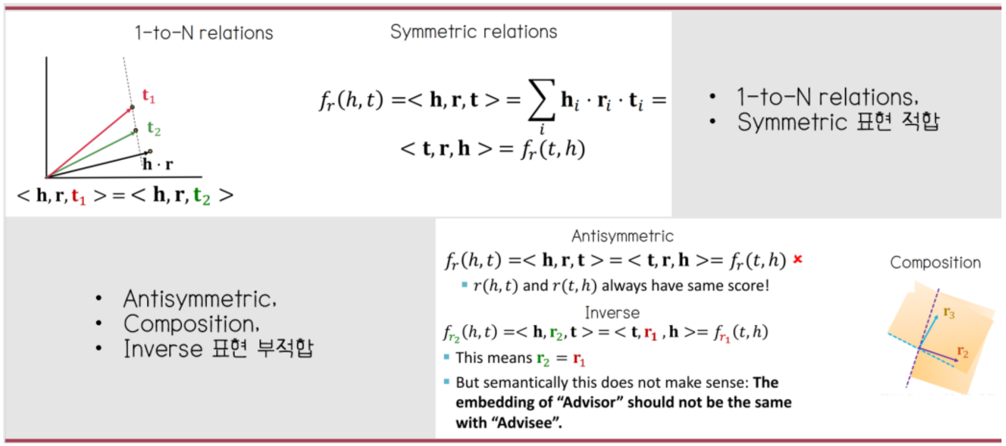




3) DistMult

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.

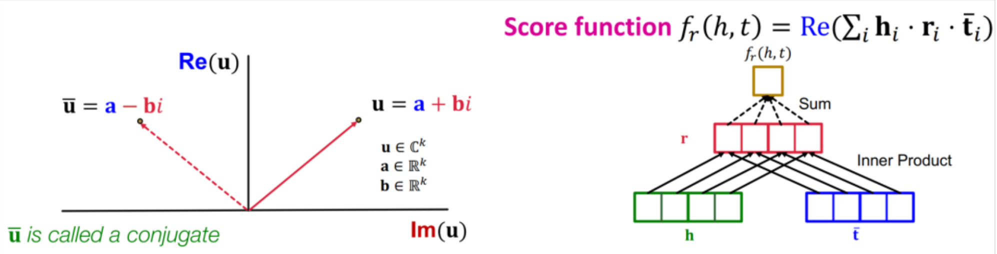




4) ComplEx

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

* function



DistMult → ComplEx

- 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	$-\lVert \mathbf{h}+\mathbf{r}-\mathbf{t} \rVert$	$\mathbf{h},\mathbf{t},\mathbf{r} \in \mathbb{R}^k$	×	✓	✓	✓	×
TransR	$-\lVert \mathbf{W}_r\mathbf{h}+\mathbf{r} - \mathbf{W}_r\mathbf{t} \rVert$	$\mathbf{h},\mathbf{t},\mathbf{r} \in \mathbb{R}^k$, $\mathbf{W}_r \in \mathbb{R}^k$	✓	✓	✓	×	✓
DistMult	$<\mathbf{h},\mathbf{r},\mathbf{t}>$	$\mathbf{h},\mathbf{t},\mathbf{r} \in \mathbb{R}^k$	✓	×	×	×	✓
ComplEx	$\mathrm{Re}(<\mathbf{h},\mathbf{r},\bar{\mathbf{t}}>)$	$\mathbf{h},\mathbf{t},\mathbf{r} \in \mathbb{C}^k$	✓	✓	✓	×	✓