

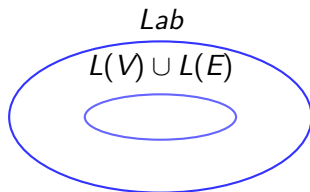
Translational Embeddings

January 10, 2022

Translating embeddings

Definition

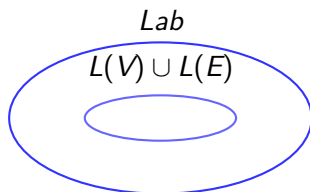
Let $KG = (V, E, L; \vdash)$ be a **knowledge graph** with a set of vertices V , a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation \vdash .
A knowledge graph embedding is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbf{R}^n$.



Translating embeddings

Definition

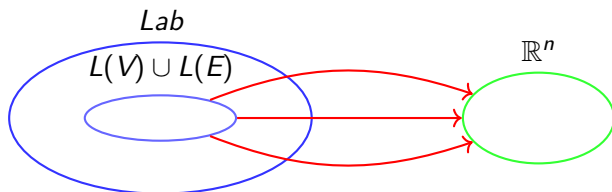
Let $KG = (V, E, L; \vdash)$ be a **knowledge graph** with a set of vertices V , a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation \vdash .
A knowledge graph embedding is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbb{R}^n$.



Translating embeddings

Definition

Let $KG = (V, E, L; \vdash)$ be a **knowledge graph** with a set of vertices V , a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation \vdash . A **knowledge graph embedding** is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbb{R}^n$.



TransE idea

TransE idea

Graph as edgelist: set of (h, ℓ, t) statements

TransE idea

Graph as edgelist: set of (h, ℓ, t) statements

Idea: $\mathbf{h} + \mathbf{\ell} \approx \mathbf{t}$ (bold symbols represent embeddings)

TransE idea

Graph as edgelist: set of (h, ℓ, t) statements

Idea: $\mathbf{h} + \ell \approx \mathbf{t}$ (bold symbols represent embeddings)

Compute *dissimilarity*: $d(\mathbf{h} + \ell, \mathbf{t}) = \|\mathbf{h} + \ell - \mathbf{t}\|$ (chose your norm, usually L2)

TransE idea

Graph as edgelist: set of (h, ℓ, t) statements

Idea: $\mathbf{h} + \ell \approx \mathbf{t}$ (bold symbols represent embeddings)

Compute *dissimilarity*: $d(\mathbf{h} + \ell, \mathbf{t}) = \|\mathbf{h} + \ell - \mathbf{t}\|$ (chose your norm, usually L2)

Minimize:

$$\mathcal{L} = \sum_{(h, \ell, t) \in S} \sum_{(h', \ell, t') \in S'_{(h, \ell, t)}} [\gamma + d(\mathbf{h} + \ell, \mathbf{t}) - d(\mathbf{h}' + \ell, \mathbf{t}')]_+$$

Objective function

$$\mathcal{L} = \sum_{(h, \ell, t) \in S} \sum_{(h', \ell, t') \in S'_{(h, \ell, t)}} [\gamma + d(\mathbf{h} + \ell, \mathbf{t}) - d(\mathbf{h}' + \ell, \mathbf{t}')]_+$$

Objective function

$$\mathcal{L} = \sum_{(h, \ell, t) \in S} \sum_{(h', \ell, t') \in S'_{(h, \ell, t)}} [\gamma + d(\mathbf{h} + \ell, \mathbf{t}) - d(\mathbf{h}' + \ell, \mathbf{t}')]_+$$

- $d(\mathbf{h} + \ell, \mathbf{t})$ is score for positive edges (or triples)

Objective function

$$\mathcal{L} = \sum_{(h, \ell, t) \in S} \sum_{(h', \ell, t') \in S'_{(h, \ell, t)}} [\gamma + d(\mathbf{h} + \ell, \mathbf{t}) - d(\mathbf{h}' + \ell, \mathbf{t}')]_+$$

- $d(\mathbf{h} + \ell, \mathbf{t})$ is score for positive edges (or triples)
- $d(\mathbf{h}' + \ell, \mathbf{t}')$ is score for negative edges (or triples)

Objective function

$$\mathcal{L} = \sum_{(h, \ell, t) \in S} \sum_{(h', \ell, t') \in S'_{(h, \ell, t)}} [\gamma + d(\mathbf{h} + \ell, \mathbf{t}) - d(\mathbf{h}' + \ell, \mathbf{t}')]_+$$

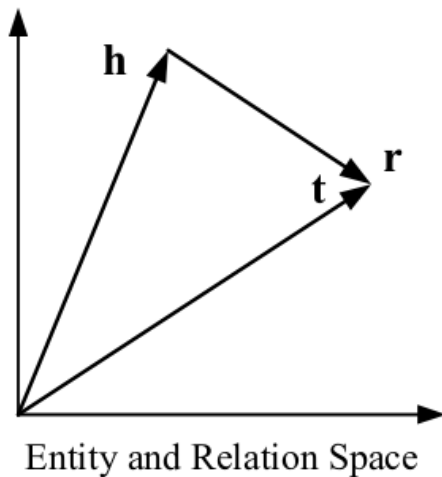
- $d(\mathbf{h} + \ell, \mathbf{t})$ is score for positive edges (or triples)
- $d(\mathbf{h}' + \ell, \mathbf{t}')$ is score for negative edges (or triples)
- γ is a margin parameter

Objective function

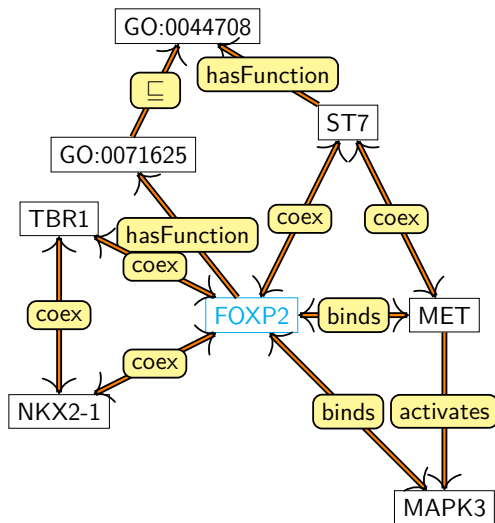
$$\mathcal{L} = \sum_{(h, \ell, t) \in S} \sum_{(h', \ell, t') \in S'_{(h, \ell, t)}} [\gamma + d(\mathbf{h} + \ell, \mathbf{t}) - d(\mathbf{h}' + \ell, \mathbf{t}')]_+$$

- $d(\mathbf{h} + \ell, \mathbf{t})$ is score for positive edges (or triples)
- $d(\mathbf{h}' + \ell, \mathbf{t}')$ is score for negative edges (or triples)
- γ is a margin parameter
- $[x]_+ = \max(0, x)$

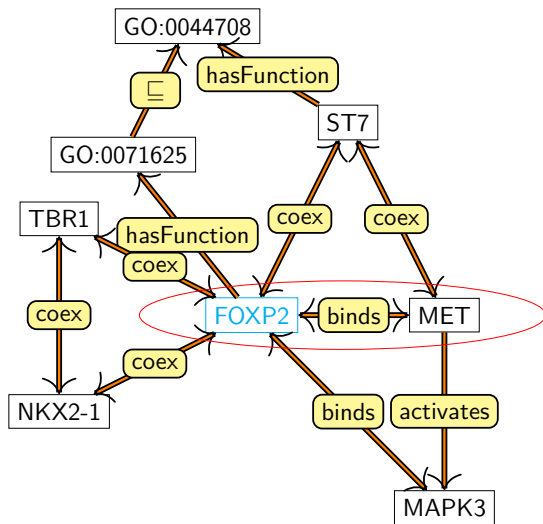
Translating embeddings



Translating embeddings

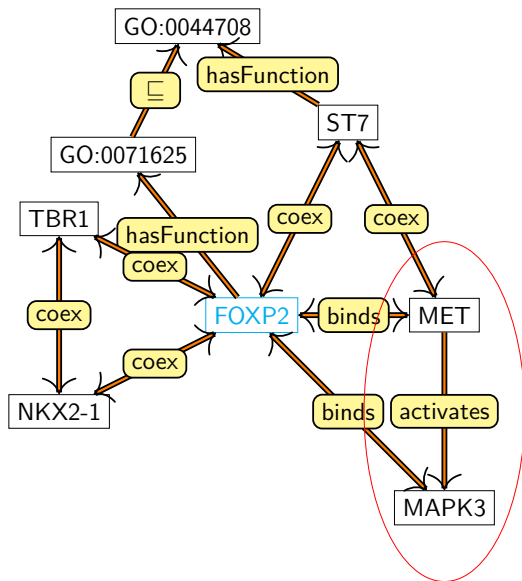


Translating embeddings



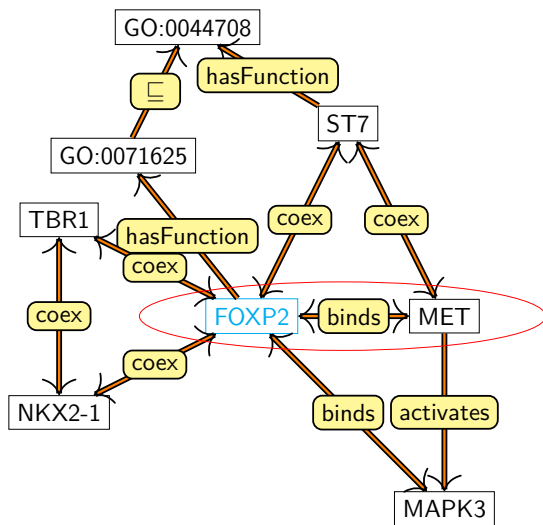
- FOXP2 + binds = MET

Translating embeddings



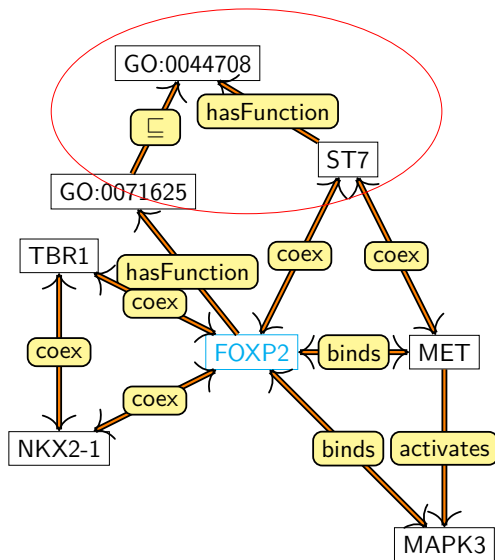
- FOXP2 + binds = MET
- MET + activates = MAPK3

Translating embeddings



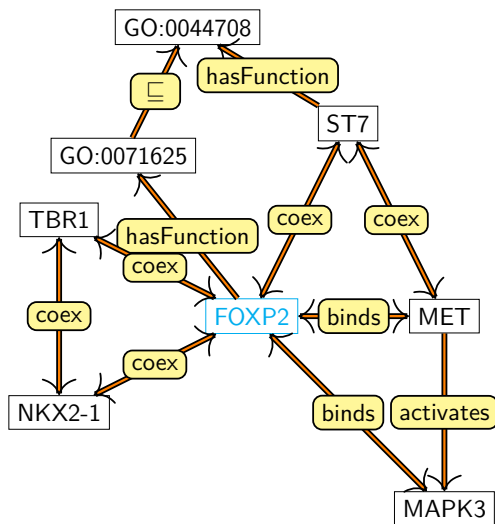
- FOXP2 + binds = MET
- MET + activates = MAPK3
- MET + binds = FOXP2

Translating embeddings



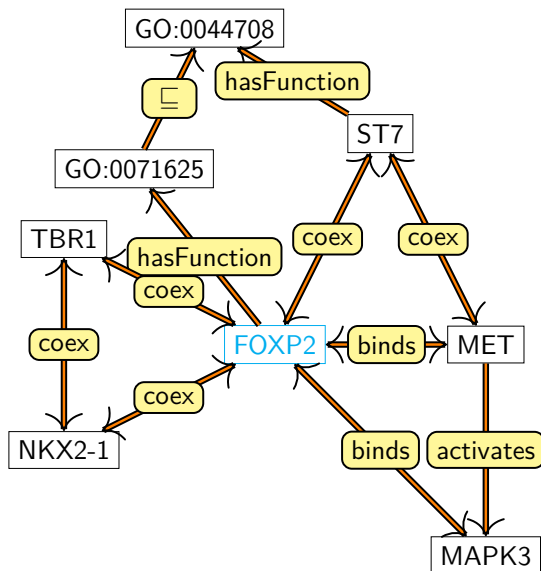
- FOXP2 + binds = MET
- MET + activates = MAPK3
- MET + binds = FOXP2
- ST7 + hasFunction = GO:0044708

Translating embeddings



- FOXP2 + binds = MET
- MET + activates = MAPK3
- MET + binds = FOXP2
- ST7 + hasFunction = GO:0044708
- ...

Translating embeddings



- $\text{FOXP2} + \text{binds} - \text{MET} = 0$
- $\text{MAP} + \text{activates} - \text{MAPK3} = 0$
- $\text{MET} + \text{binds} - \text{FOXP2} = 0$
- $\text{ST7} + \text{hasFunction} - \text{GO:0044708} = 0$
- ...

Some properties of TransE

- graph-based
 - works well on RDF graphs
 - and ontology graphs

Some properties of TransE

- graph-based
 - works well on RDF graphs
 - and ontology graphs
- 1:1 relations only
 - not suitable for hierarchies (1-N relations)
 - not suitable for N-N relations
 - no transitive, symmetric, reflexive relations

Translating embeddings

TransH deals with the 1-N, N-1 and N-N relationships

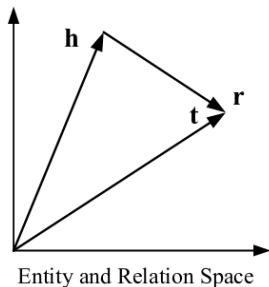


Figure: TransE representation

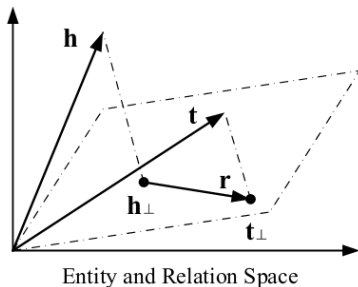
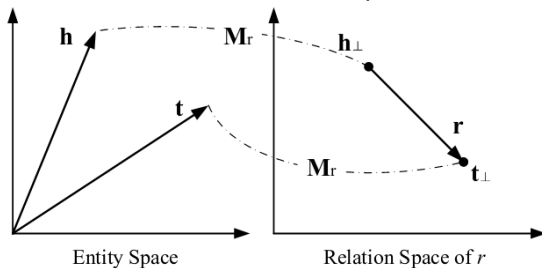


Figure: TransH representation

Translating embeddings

TransR: each relation has its own semantic space.



(c) TransR.

- Python package to generate knowledge graph embeddings
- supports many different graph embedding types: TransE, TransH, TransR, TransD, RESCAL, etc.
- mOWL integration

The End