Graph Neural Networks

Lecture 9

Danila Biktimirov
Applied Computer Science
Neapolis University Pafos

22 April 2025

Overview

TransE Learning Algorithm

Algorithm 1 Learning TransE

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

initialize $\boldsymbol{\ell} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$

$\boldsymbol{\ell} \leftarrow \boldsymbol{\ell} / \|\boldsymbol{\ell}\|$ for each $\ell \in L$

$\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

Entities and relations are initialized uniformly, and normalized

loop

$\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$

$S_{batch} \leftarrow \text{sample}(S, b)$ // sample a minibatch of size b

$T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets

for $(h, \ell, t) \in S_{batch}$ **do**

$(h', \ell, t') \leftarrow \text{sample}(S'_{(h,\ell,t)})$ // sample a corrupted triplet

Negative sampling with triplet that does not appear in the KG

$T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell, t'))\}$

end for

Update embeddings w.r.t.

$$\sum_{((h,\ell,t),(h',\ell,t')) \in T_{batch}} \nabla \big[\gamma + \underbrace{d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t})}_{\text{positive sample}} - \underbrace{d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})}_{\text{negative sample}}\big]_+$$

d represents distance (negative of score)

end loop

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

Relation Patterns

- (Anti)Symmetry: $r(h,t) \Rightarrow r(t,h) \;\; (\; r(h,t) \Rightarrow \neg r(t,h) \;) \quad \forall h,t$
- Inversion: $r_2(h,t) \Rightarrow r_1(t,h)$
- Transitivity: $r_1(x,y) \wedge r_2(y,z) \Rightarrow r_3(x,z) \quad \forall x,y,z$
- 1-to-N: $r(h,t_1), r(h,t_2), \ldots, r(h,t_n)$ are all True

Today: Reasoning over KGs

- **Goal:**
  - How to perform multi-hop reasoning over KGs?
- **Reasoning over Knowledge Graphs**
  - Answering multi-hop queries
    - Path Queries
    - Conjunctive Queries
  - Query2Box

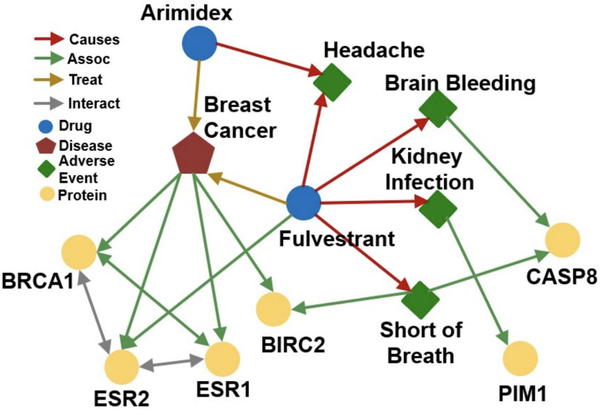
“Queries” in Knowledge Graphs

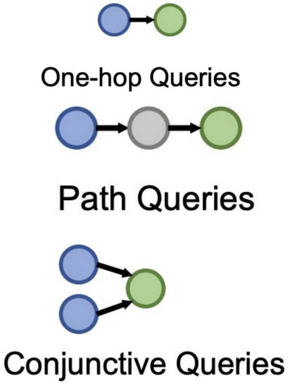
Multi-hop Queries

Can we use knowledge graphs to work not only with triples,
but also with long chains of relations?

Yes, we can.

Types of Queries





One-hop

- **KG completion**: Is link (h, r, t) in the KG?

$$\updownarrow$$

- **One-hop query**: Is t an answer to query (h, r)?

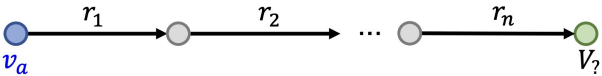
Path

- An n-hop path query q can be represented by

$$q = (v_a, (r_1, \ldots, r_n))$$

- v_a is an "anchor" entity
- Let answers to q in graph G be denoted by q_G

Query Plan of q:

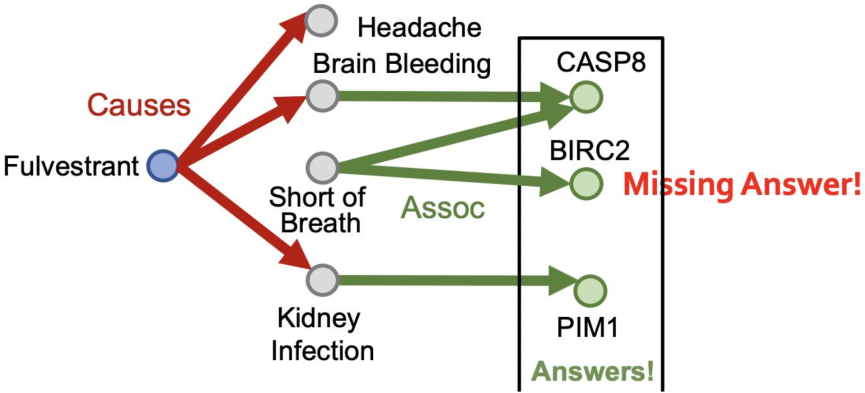


How to Answer

Intuitively – traversal

Problem – lots of missing data

Example of Missing Data Problem

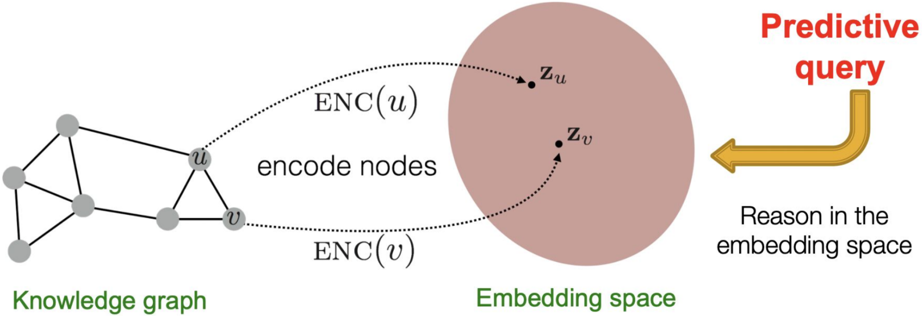


Using Completion

Using the knowledge graph completion approach doesn't help much – the graph becomes dense, and traversal complexity becomes exponential.

What to Do

Let's recall embeddings

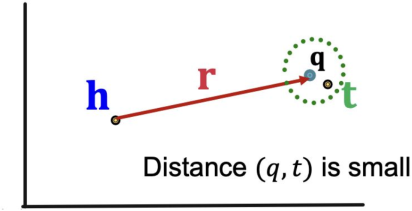


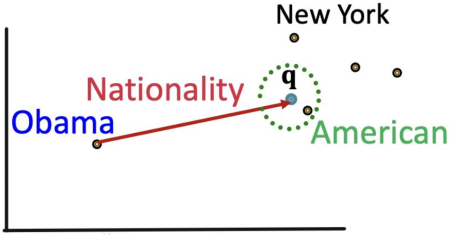
We will use TransE

Another way to interpret this is that:

- Query embedding: $\mathbf{q} = \mathbf{h} + \mathbf{r}$
- Goal: query embedding $\mathbf{q}$ is close to the answer embedding $\mathbf{t}$

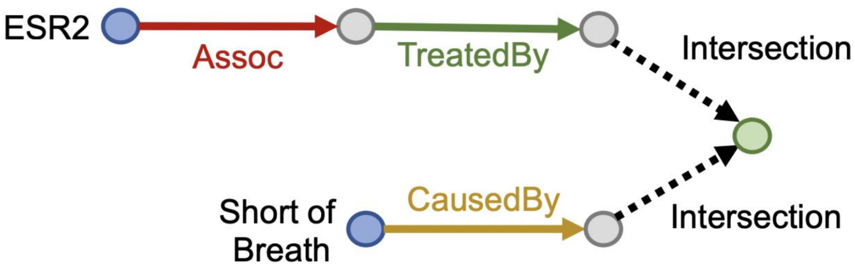
$$f_q(t) = -\|\mathbf{q} - \mathbf{t}\|$$



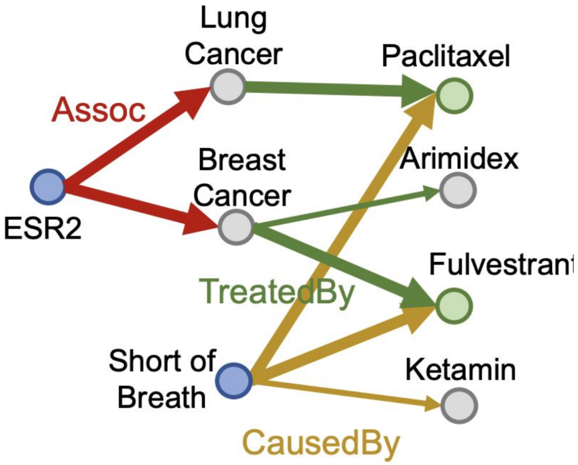


Conjunctive

Query plan:

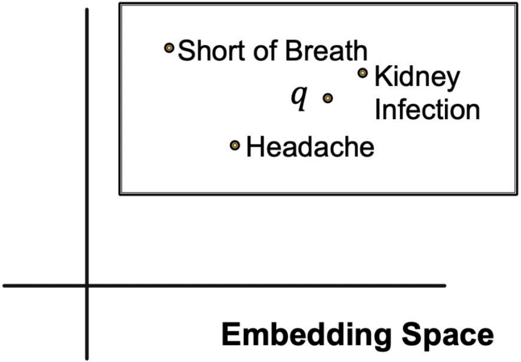


Query Example



Box Embeddings

$$\mathbf{q} = (\mathit{Center}(q), \mathit{Offset}(q))$$



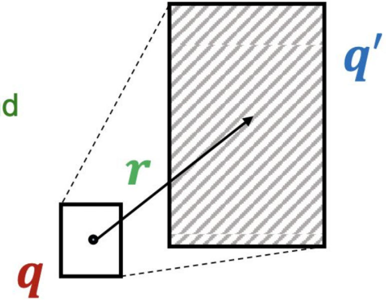
For example, we can embed the adverse events of Fulvestrant with **a box that enclose all the answer entities**.

Projection Operator

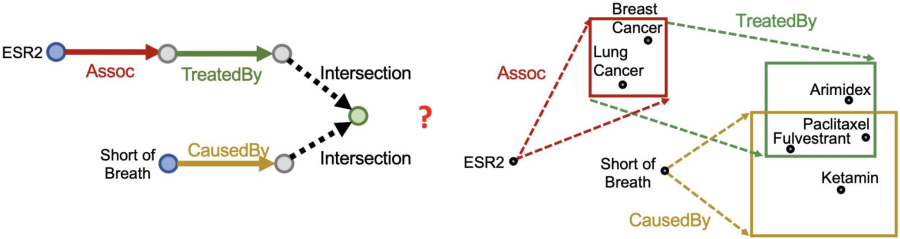
- **P**: Box $\times$ Relation $\to$ Box

$$\mathrm{Cen}(q') = \mathrm{Cen}(q) + \mathrm{Cen}(r)$$
$$\mathrm{Off}(q') = \mathrm{Off}(q) + \mathrm{Off}(r)$$

"×" (cross) means the projection operator is a **relation** from any box and **relation** to a new box



Working Example



Intersection Operator

- $\mathcal{I} : \mathsf{Box} \times \cdots \times \mathsf{Box} \to \mathsf{Box}$

Intersection Operator

- $\mathcal{I} : \text{Box} \times \cdots \times \text{Box} \to \text{Box}$

$$\text{Cen}(q_{\text{inter}}) = \sum_i \mathbf{w}_i \odot \text{Cen}(q_i)$$

$$\mathbf{w}_i = \frac{\exp(f_{\text{cen}}(\text{Cen}(q_i)))}{\sum_j \exp(f_{\text{cen}}(\text{Cen}(q_j)))}$$

$$\text{Cen}(q_i) \in \mathbb{R}^d \qquad \mathbf{w}_i \in \mathbb{R}^d$$

Intersection Operator

$$\mathcal{I} : \mathsf{Box} \times \cdots \times \mathsf{Box} \rightarrow \mathsf{Box}$$

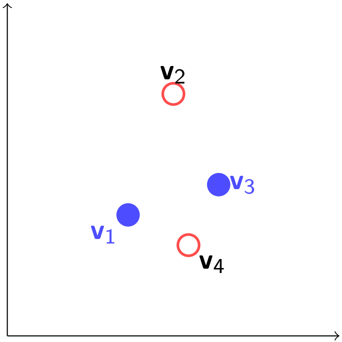
$$\mathsf{Off}(q_{\mathsf{inter}}) = \min\left(\mathsf{Off}(q_1), \ldots, \mathsf{Off}(q_n)\right) \circledast \sigma\left(f_{\mathsf{off}}\left(\mathsf{Off}(q_1), \ldots, \mathsf{Off}(q_n)\right)\right)$$

Box Distance Function

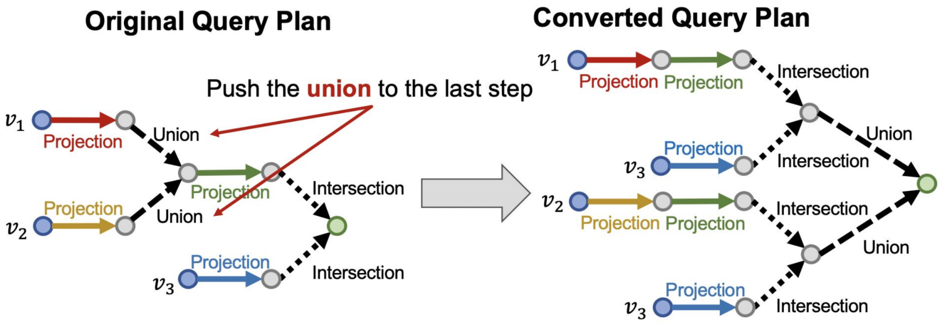
$$d_{\mathsf{box}}(\mathbf{q}, \mathbf{v}) = d_{\mathsf{out}}(\mathbf{q}, \mathbf{v}) + \alpha \cdot d_{\mathsf{in}}(\mathbf{q}, \mathbf{v})$$

$$f_{\mathbf{q}}(\mathbf{v}) = -d_{\mathsf{box}}(\mathbf{q}, \mathbf{v})$$

Union



Union



Original Query Plan
Converted Query Plan
Push the union to the last step
v_1
Projection
Union
Projection
Projection
Union
v_2
Intersection
Projection
v_3
Intersection
v_1
Projection Projection
Intersection
Union
Projection
v_3
Intersection
v_2
Intersection
Projection Projection
Union
Projection
v_3
Intersection

Distance for AND-OR Queries

- **Distance** between entity embedding and a DNF $q = q_1 \vee q_2 \vee \cdots \vee q_m$ is defined as:

$$d_{\text{box}}(\mathbf{q}, \mathbf{v}) = \min(d_{\text{box}}(\mathbf{q}_1, \mathbf{v}), \ldots, d_{\text{box}}(\mathbf{q}_m, \mathbf{v}))$$

- **The process of embedding any AND-OR query** q
  1. Transform q to equivalent DNF $q_1 \vee \cdots \vee q_m$
  2. Embed q_1 to q_m
  3. Calculate the (box) distance $d_{\text{box}}(\mathbf{q}_i, \mathbf{v})$
  4. Take the minimum of all distances
  5. **The final score** $f_q(\mathbf{v}) = -d_{\text{box}}(\mathbf{q}, \mathbf{v})$

Training

1. Randomly sample a query q from the training graph G_{train}, answer $v \in q_{G_{\text{train}}}$, and a negative sample $v' \notin q_{G_{\text{train}}}$.
    - Negative sample: Entity of same type as v but not answer.
2. Embed the query $\mathbf{q}$.
3. Calculate the score $f_q(v)$ and $f_q(v')$.
4. Optimize the loss ℓ to maximize $f_q(v)$ while minimizing $f_q(v')$:

$$\ell = -\log \sigma\big(f_q(v)\big) - \log\big(1 - \sigma\big(f_q(v')\big)\big)$$