Reasoning over Knowledge Graphs

CS224W: Machine Learning with Graphs
Jure Leskovec, Hongyu Ren, Stanford University
http://cs224w.stanford.edu

Outline of Today's Lecture

1. **Introduction to Knowledge Graphs**

2. **Knowledge Graph completion**

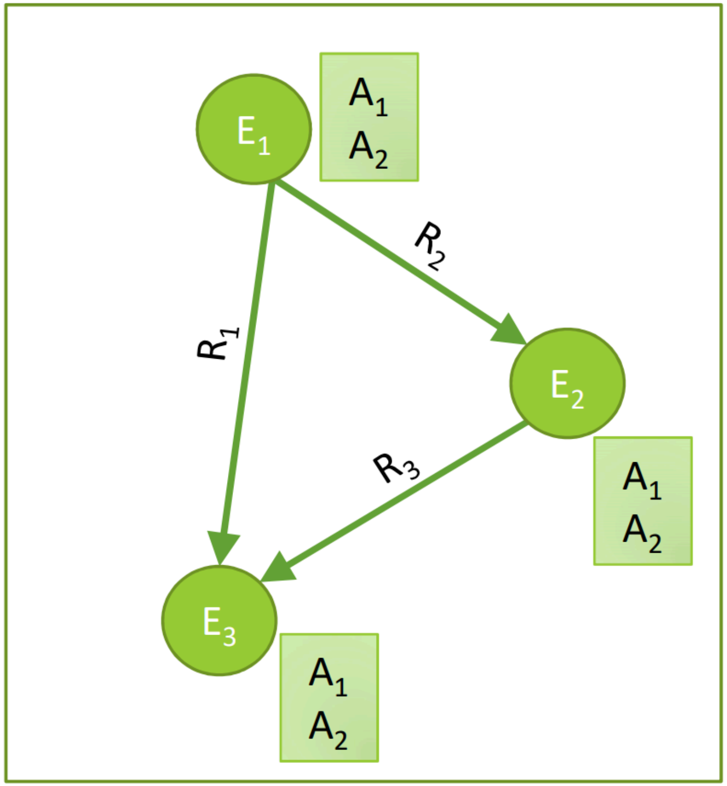
3. **Path Queries**

4. **Conjunctive Queries**

5. **Query2Box: Reasoning with Box Embeddings**

Knowledge Graphs

- Knowledge in graph form
  - Capture entities, types, and relationships
- Nodes are **entities**
- Nodes are labeled with their **types**
- Edges between two nodes capture **relationships** between entities

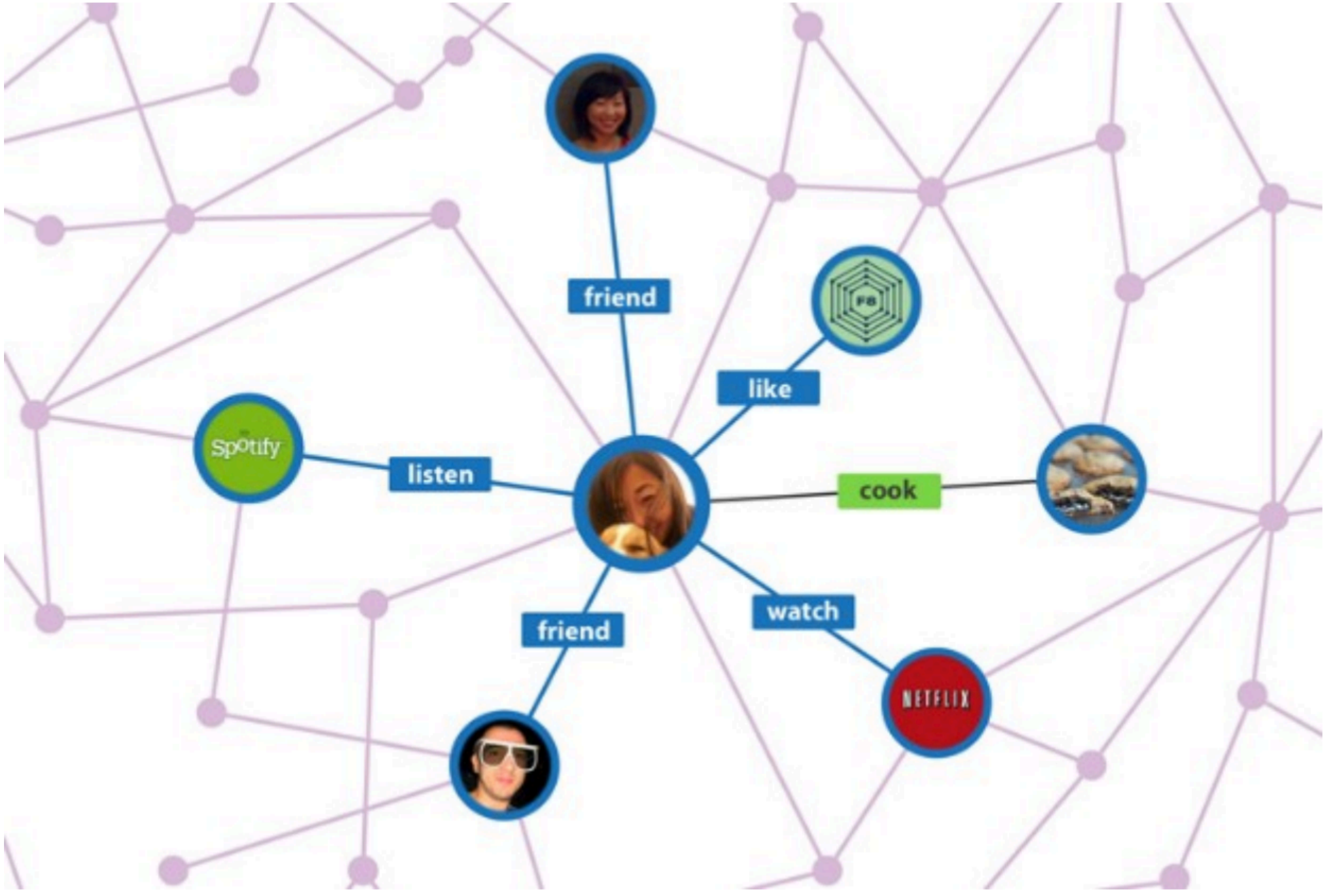


Example: Bibliographic networks

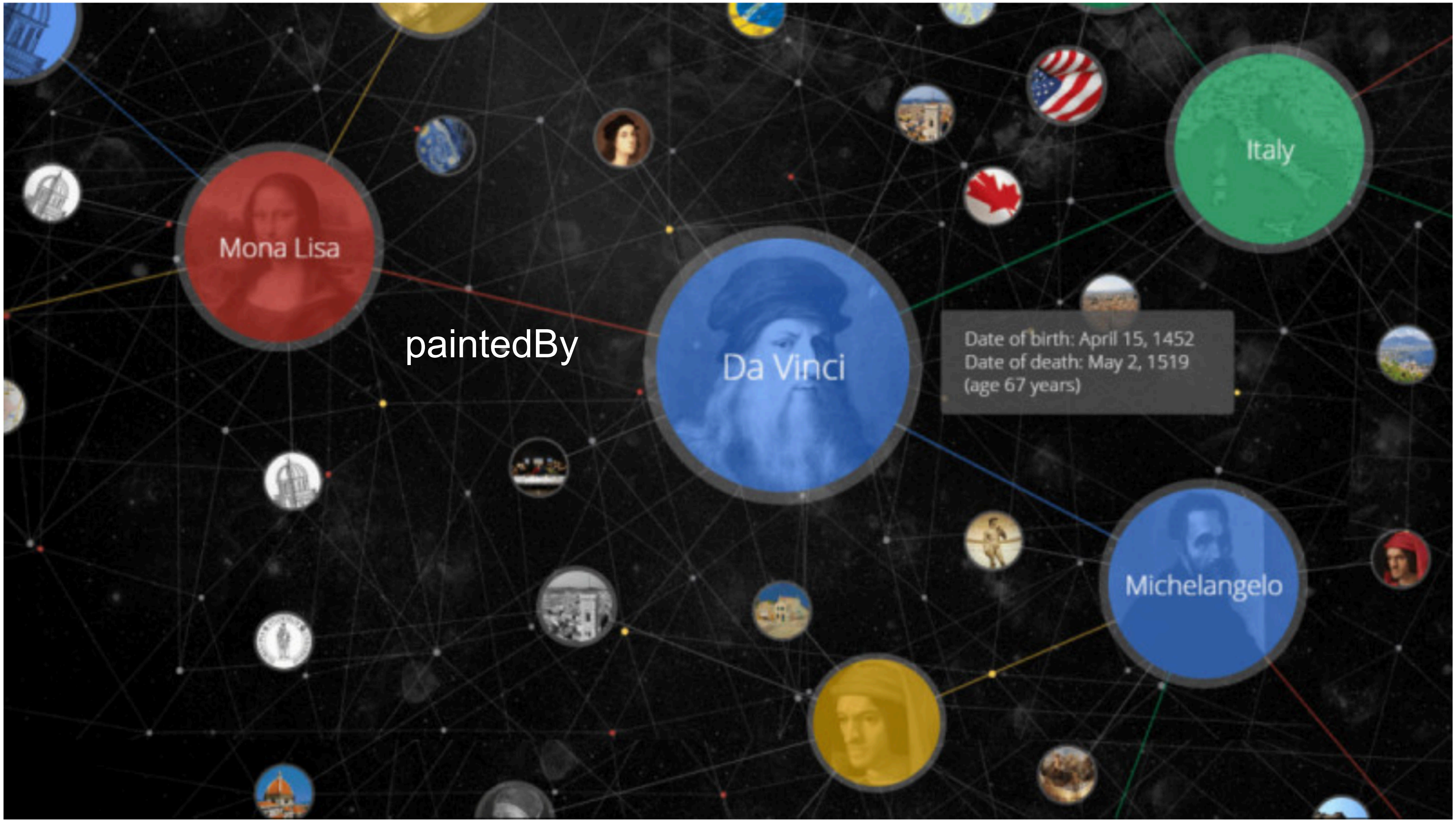
- **Node types**: paper, title, author, conference, year
- **Relation types**: pubWhere, pubYear, hasTitle, hasAuthor, cite

Example: Social networks

- **Node types**: account, song, post, food, channel
- **Relation types**: friend, like, cook, watch, listen

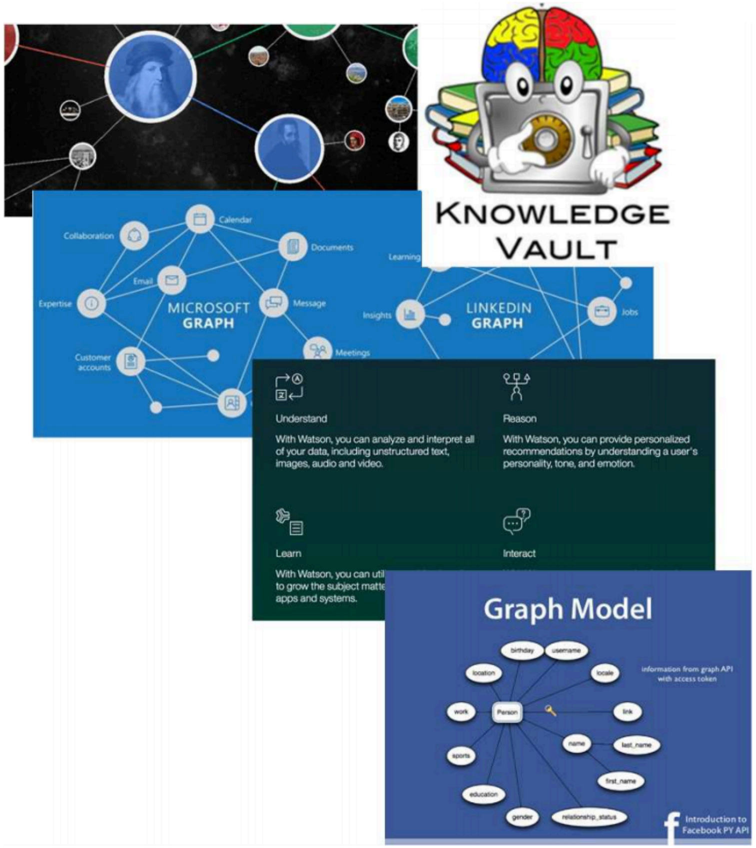


Example: Google Knowledge Graph



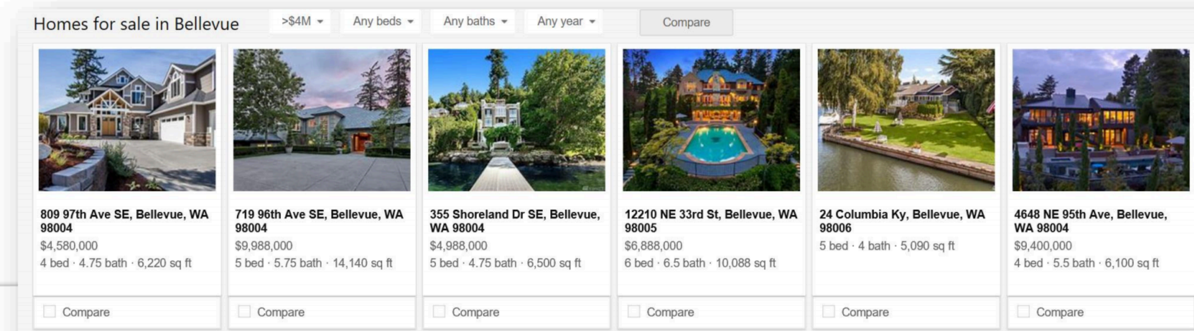
Knowledge Graphs in Practice

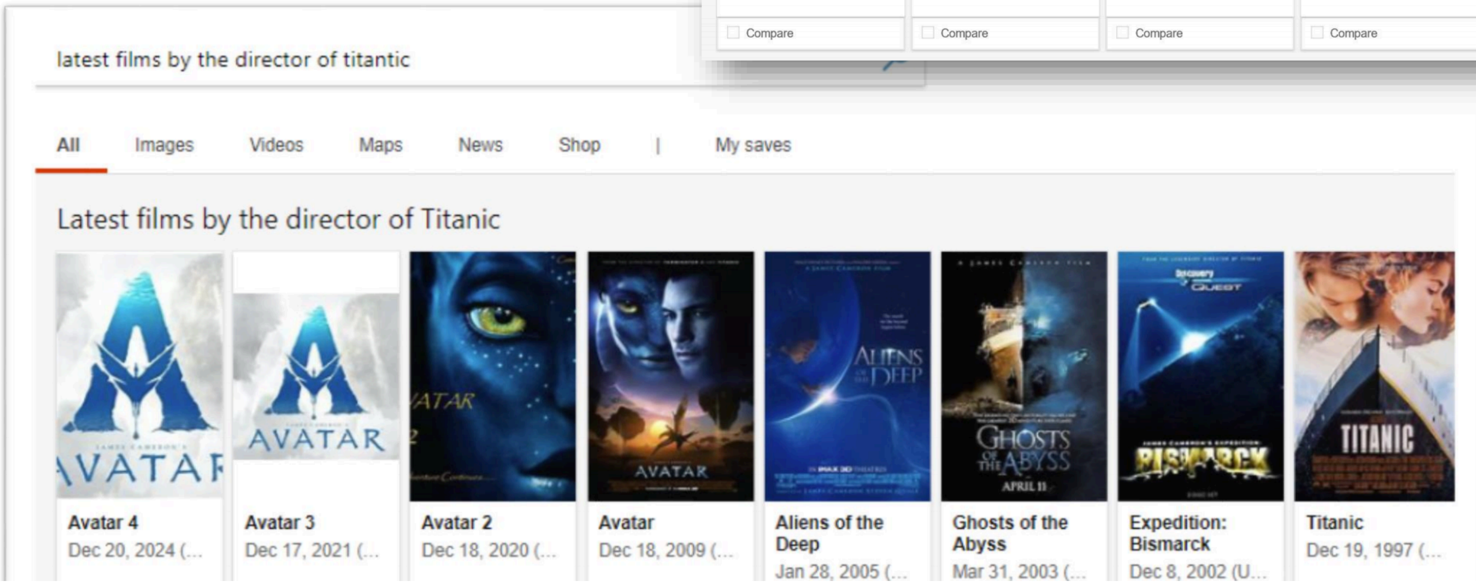
- Google Knowledge Graph
- Amazon Product Graph
- Facebook Graph API
- IBM Watson
- Microsoft Satori
- Project Hanover/Literome
- LinkedIn Knowledge Graph
- Yandex Object Answer



Applications of Knowledge Graphs

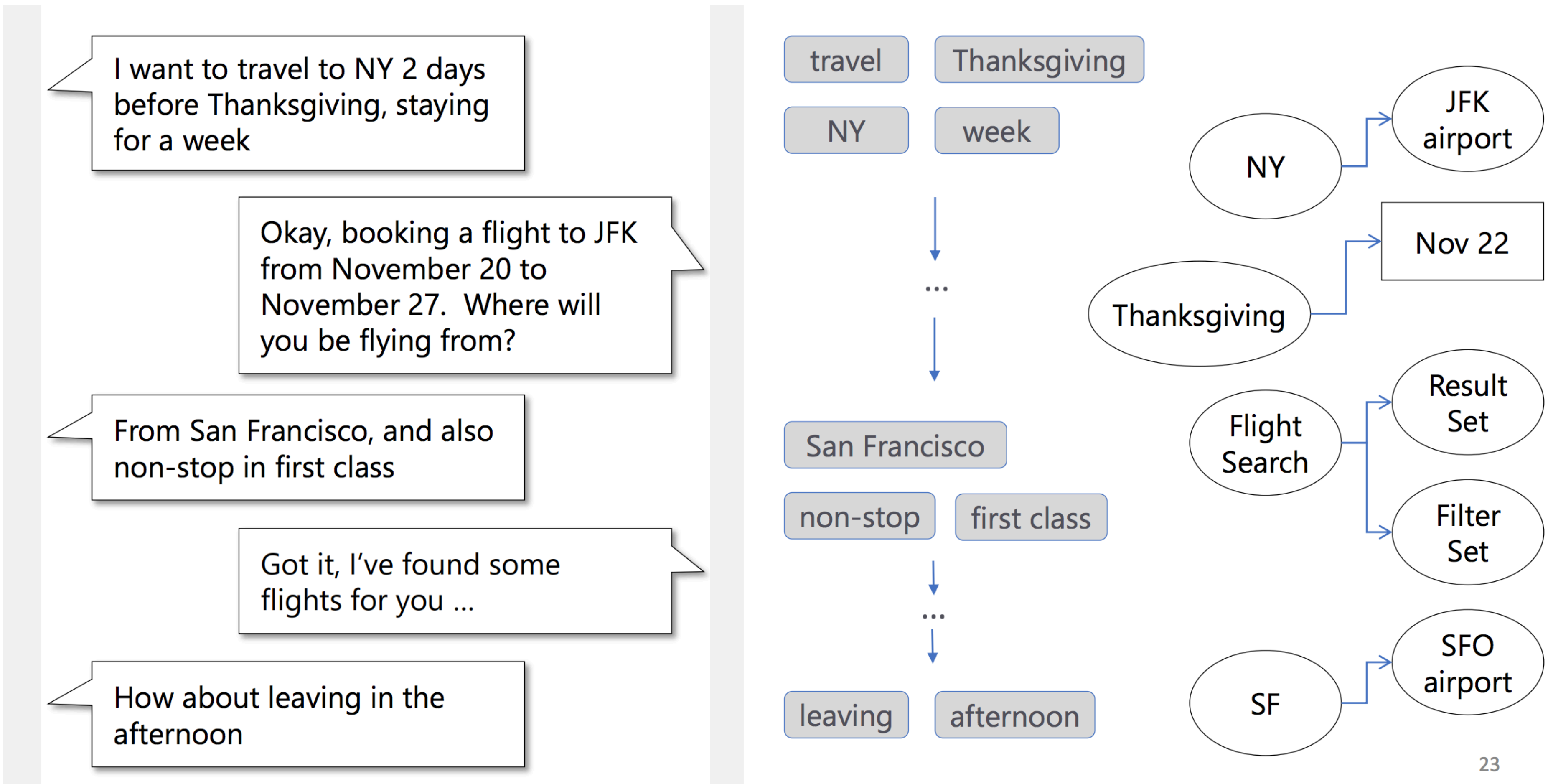
- **Serving information**



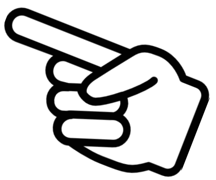


Applications of Knowledge Graphs

- Question answering and conversation agents



Outline

1. **Introduction to Knowledge Graphs**
2. **Knowledge Graph completion** 
3. **Path Queries**
4. **Conjunctive Queries**
5. **Query2Box: Reasoning with Box Embeddings**

Knowledge Graph Datasets

- Publicly available KGs:
  - FreeBase, Wikidata, Dbpedia, YAGO, NELL, etc.
- Common characteristics:
  - **Massive**: millions of nodes and edges
  - **Incomplete**: many true edges are missing

Given a massive KG, enumerating all the possible facts is intractable! ⇨ **Can we predict plausible BUT missing links?**

Example: Freebase

- Freebase
  - ~50 million **entities**
  - ~38K **relation types**
  - ~3 billion **facts/triples**

93.8% of persons from Freebase have no place of birth and 78.5% have no nationality!

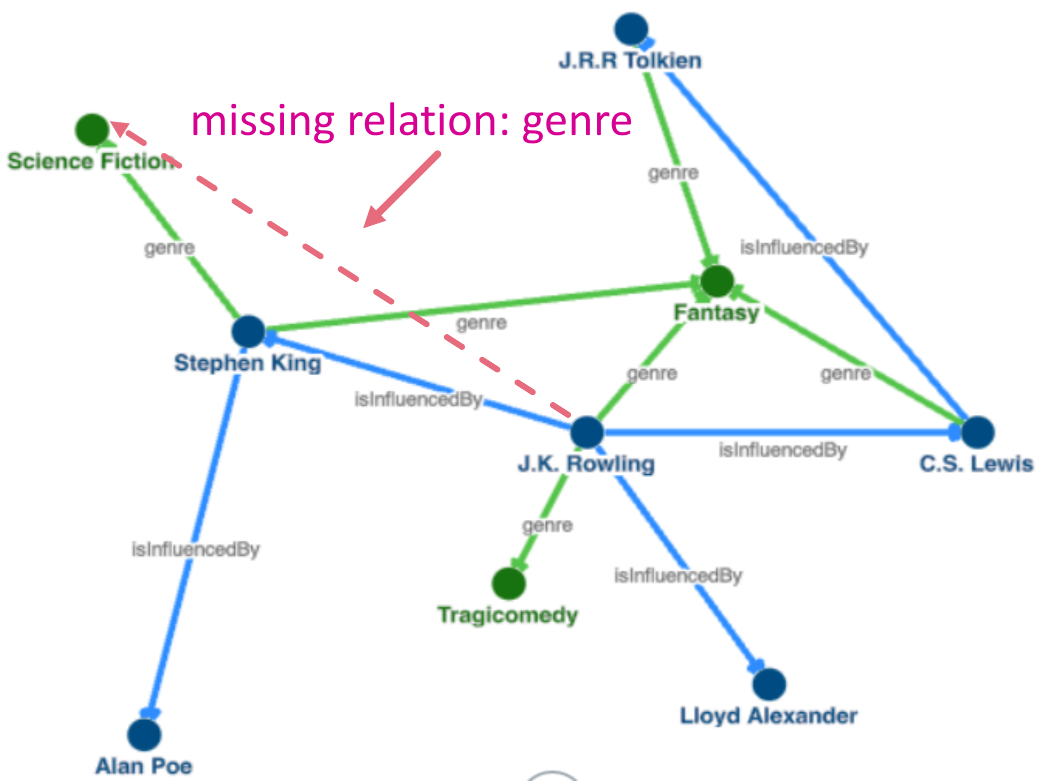
- FB15k/FB15k-237
  - A **complete** subset of Freebase, used by researchers to learn KG models

Dataset	Entities	Relations	Total Edges
FB15k	14,951	1,345	592,213
FB15k-237	14,505	237	310,079

[1] Paulheim, Heiko. "Knowledge graph refinement: A survey of approaches and evaluation methods." *Semantic web* 8.3 (2017): 489-508.
[2] Min, Bonan, et al. "Distant supervision for relation extraction with an incomplete knowledge base." *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013.

KG Completion

- Given an enormous KG, can we complete the KG / predict missing relations?
  - links + type



KG Representation

- Edges in KG are represented as **triples** (h, r, t)
  - head (h) has relation (r) with tail (t).
- Key Idea:
  - Model entities and relations in the embedding/vector space $\mathbb{R}^d$.
  - Given a true triple (h, r, t), the goal is that the embedding of (h, r) **should be close** to the embedding of t.
    - How to embed (h, r)?
    - How to define closeness?

Relation Patterns

- Symmetric Relations:
$$r(h,t) \Rightarrow r(t,h) \quad \forall h,t$$
  - **Example**: Family, Roommate
- Composition Relations:
$$r_1(x,y) \wedge r_2(y,z) \Rightarrow r_3(x,z) \quad \forall x,y,z$$
  - **Example**: My mother's husband is my father.
- 1-to-N, N-to-1 relations:
$r(h,t_1), r(h,t_2), \ldots, r(h,t_n)$ are all True.
  - **Example**: r is "StudentsOf"

TransE

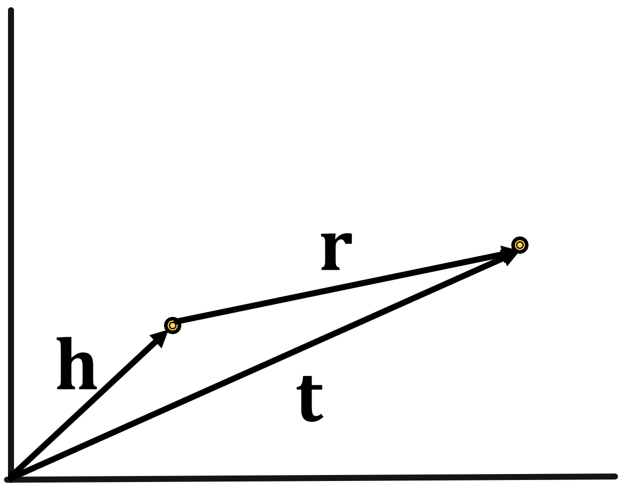
- **Translation Intuition**:

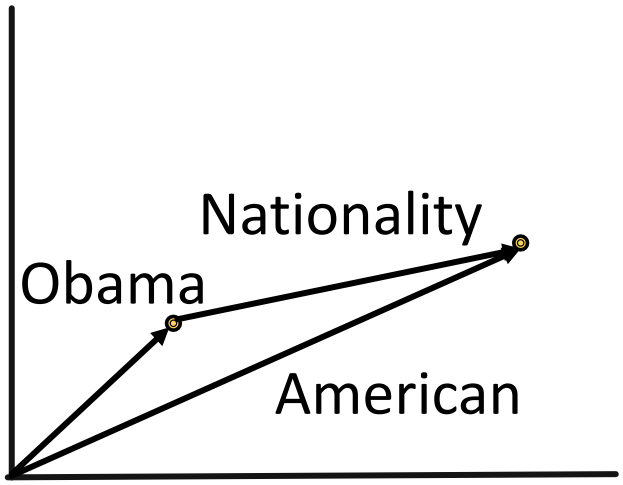
For a triple (h, r, t), $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$,

$$\mathbf{h} + \mathbf{r} = \mathbf{t}$$

NOTATION: embedding vectors will appear in boldface

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





Bordes, Antoine, et al. "Translating embeddings for modeling multi-relational data." *Advances in neural information processing systems*. 2013.

TransE Training

- Translation Intuition: for a triple (h, r, t),

$$\mathbf{h} + \mathbf{r} = \mathbf{t}$$

Max margin loss:

$$\mathcal{L} = \sum_{(h,r,t)\in G,(h,r,t')\notin G} [\gamma + \underbrace{f_r(h,t)}_{\text{Valid triple}} - \underbrace{f_r(h,t')}_{\text{Corrupted triple}}]_+$$

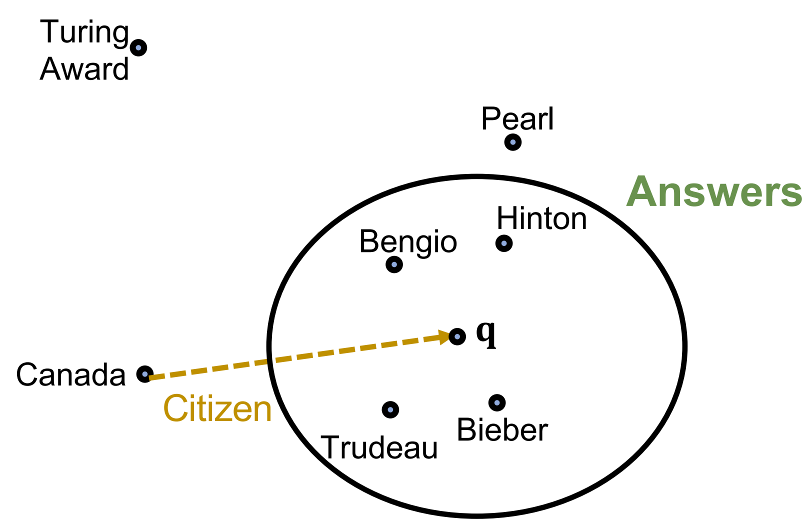
where γ is the margin, i.e., the smallest distance tolerated by the model between a valid triple and a corrupted one.

NOTE: check lecture 7 for a more in-depth discussion of TransE!

Link Prediction in a KG using TransE

- Who has won the Turing award?

- Who is a Canadian citizen?



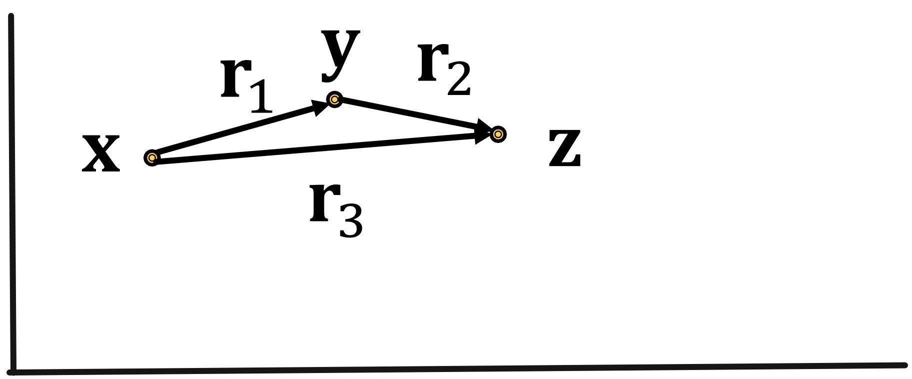
Composition in TransE

- Composition Relations:

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

- **Example**: My mother's husband is my father.
- In TransE:

  $$r_3 = r_1 + r_2 \checkmark$$



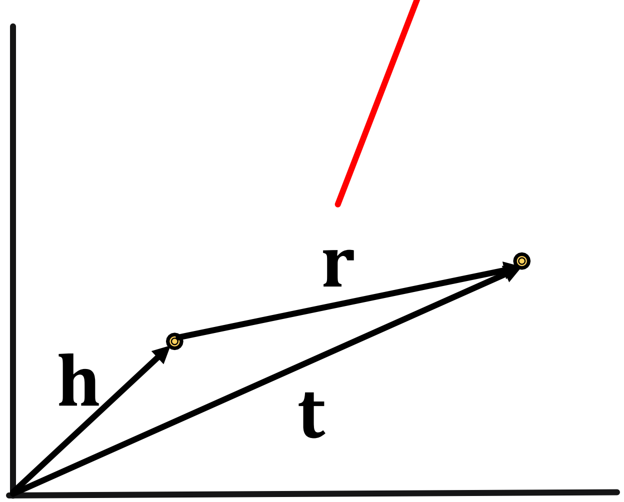
Limitation: Symmetric Relations

- Symmetric Relations:

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

- **Example**: Family, Roommate
- In TransE:

$$r = 0,\ h = t$$ ✗



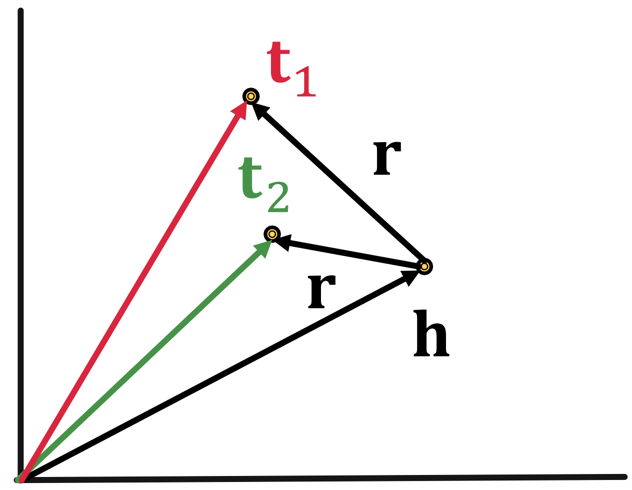
If we want TransE to handle symmetric relations r, for all h,t that satisfy $r(h,t)$, $r(t,h)$ is also True, which means $\|h+r-t\|=0$ and $\|t+r-h\|=0$. Then $r=0$ and $h=t$, however h and t are two different entities and should be mapped to different locations.

Limitation: N-ary Relations

- 1-to-N, N-to-1, N-to-N relations.
- **Example**: (h, r, t_1) and (h, r, t_2) both exist in the knowledge graph, e.g., r is "StudentsOf"

With TransE, t_1 and t_2 will map to the same vector, although they are different entities.

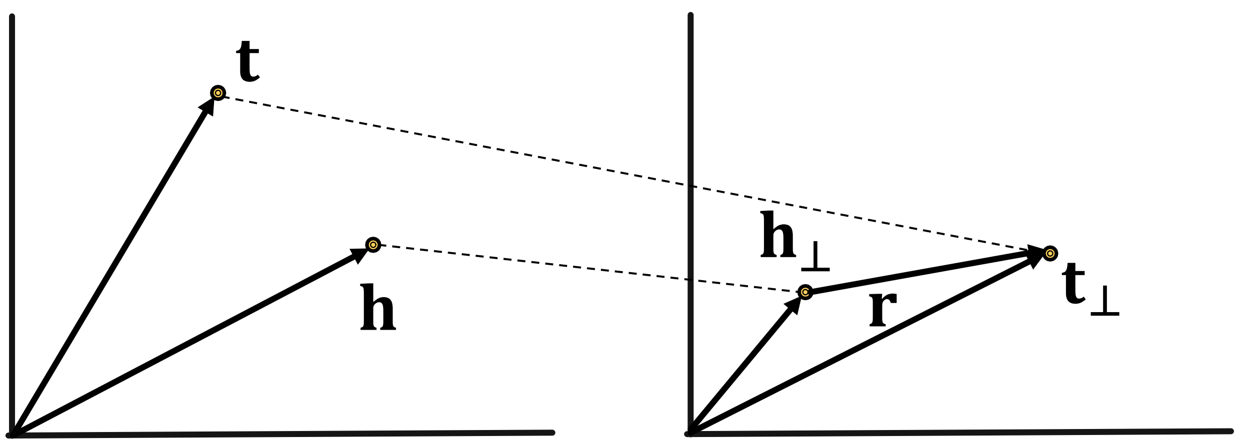
- $\mathbf{t}_1 = \mathbf{h} + \mathbf{r} = \mathbf{t}_2$
- $\mathbf{t}_1 \neq \mathbf{t}_2$ contradictory!



TransR

- TransR: model entities as vectors in the entity space $\mathbb{R}^d$ and **model each relation as vector r in relation space** $\mathbb{R}^k$ with $\mathbf{M}_r \in \mathbb{R}^{k\times d}$ as the projection matrix.

- $h_\perp = M_r h, \ t_\perp = M_r t$
- $f_r(h,t) = ||h_\perp + r - t_\perp||$



Lin, Yankai, et al. "Learning entity and relation embeddings for knowledge graph completion." *AAAI*. 2015.

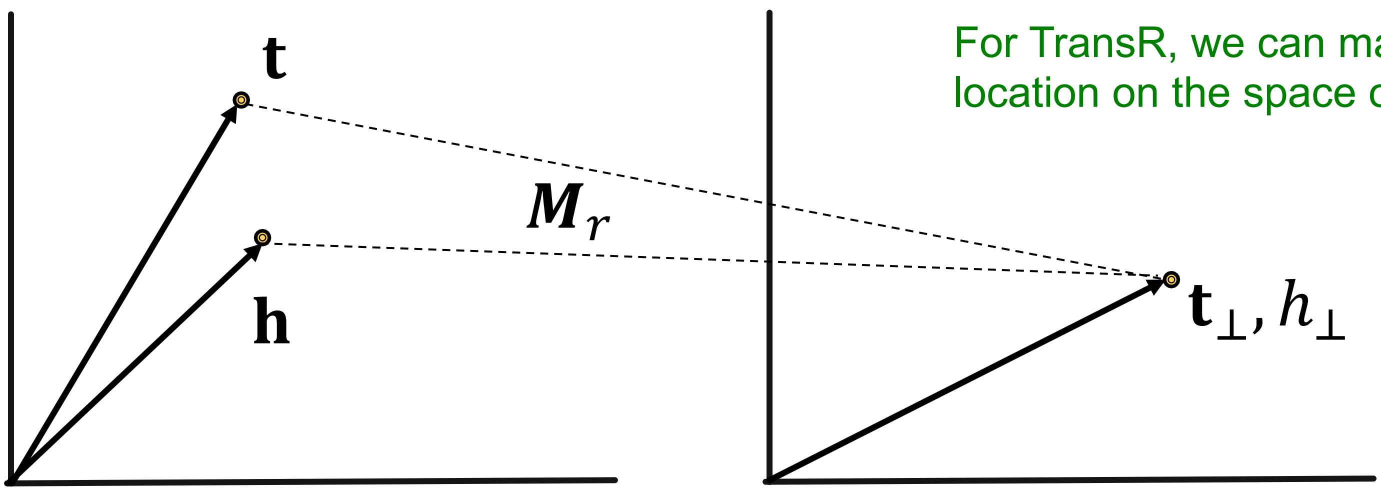
Symmetric Relations in TransR

- Symmetric Relations:

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

- **Example**: Family, Roommate

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

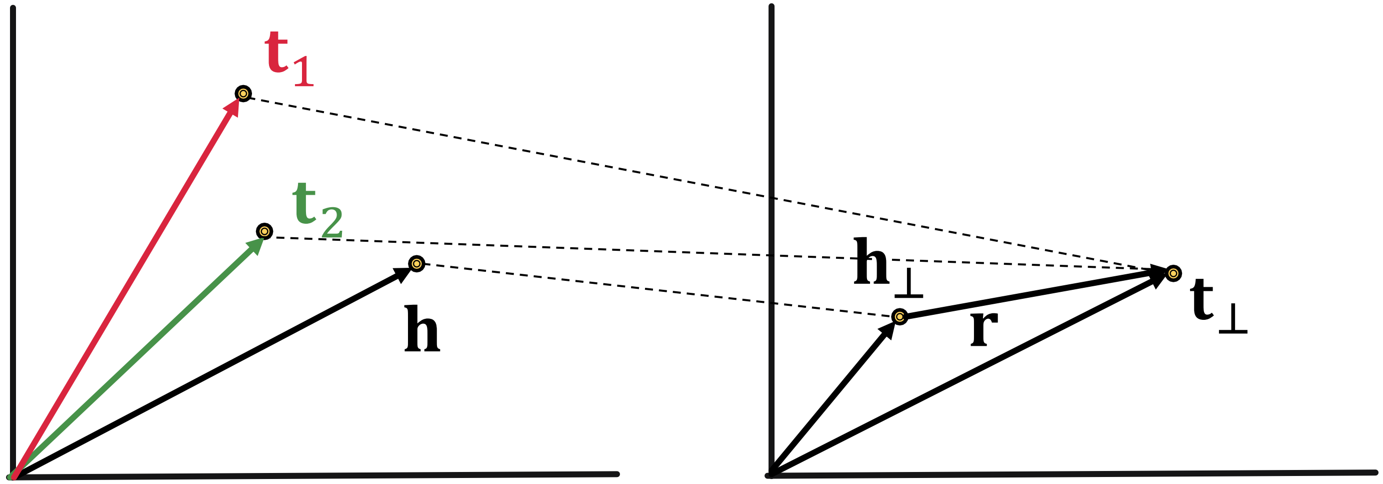


For TransR, we can map h and t to the same location on the space of relation r.

N-ary Relations in TransR

- 1-to-N, N-to-1, N-to-N relations
- **Example**: If (h, r, t_1) and (h, r, t_2) exist in the knowledge graph.

We can learn M_r so that $t_\perp = M_r t_1 = M_r t_2$, note that t_1 does not need to be equal to t_2!



Limitation: Composition in TransR

- Composition Relations:
  $$r_1(x,y) \wedge r_2(y,z) \Rightarrow r_3(x,z) \quad \forall x,y,z$$
- **Example**: My mother's husband is my father.

Each relation has different space.
It is **not naturally compositional** for multiple relations! ✘

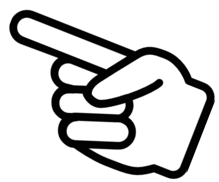
Translation-Based Embedding

Embedding	Entity	Relation	$f_r(h,t)$
TransE	$h, t \in \mathbb{R}^d$	$r \in \mathbb{R}^d$	$\|\|h + r - t\|\|$
TransR	$h, t \in \mathbb{R}^d$	$r \in \mathbb{R}^k, M_r \in \mathbb{R}^{k \times d}$	$\|\|M_r h + r - M_r t\|\|$

Embedding	Symmetry	Composition	One-to-many
TransE	✘	✓	✘
TransR	✓	✘	✓

Outline

1. **Introduction to Knowledge Graphs**
2. **Knowledge Graph completion**
3. **Path Queries**



4. **Conjunctive Queries**
5. **Query2Box: Reasoning with Box Embeddings**

Query Types on KG

- Can we do multi-hop reasoning, i.e., answer complex queries **efficiently** on an incomplete, massive KG?

Query Types	Examples
One-hop Queries	Where did Hinton graduate?
Path Queries	Where did Turing Award winners graduate?
Conjunctive Queries	Where did Canadians with Turing Award graduate?
EPFO Queries	Where did Canadians with Turing Award or Nobel graduate?

One-hop Queries

- We can formulate link prediction problems as answering one-hop queries.

- Link prediction: Is link (h, r, t) True?

$$\Updownarrow$$

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

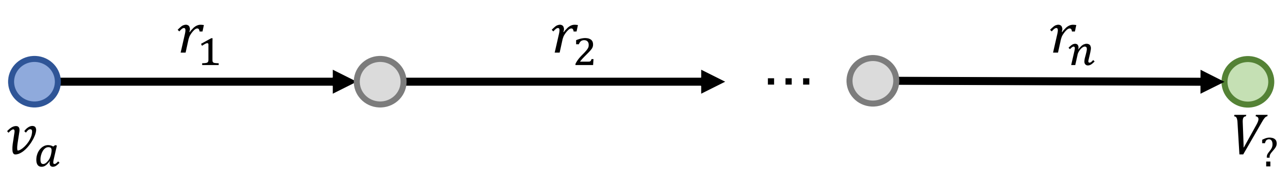
Path Queries

- Generalize one-hop queries to path queries by adding more relations on the path.
- Path queries can be represented by

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

v_a is a constant node, answers are denoted by $[\![q]\!]$.

Computation graph of q:

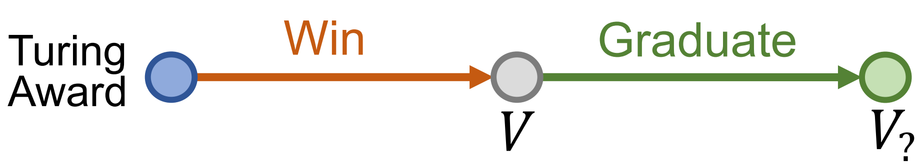


Computation graph of path queries is a chain.

Path Queries

"Where did Turing Award winners graduate?"

- v_a is "Turing Award".
- (r_1, r_2) is ("win", "graduate").



Given a KG, how to answer the query?

Traversing Knowledge Graphs

- Answer path queries by traversing the KG.

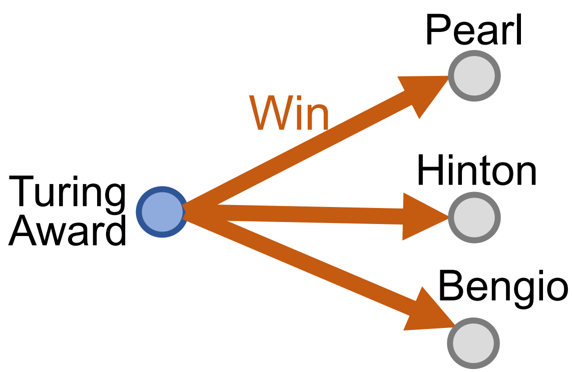
"Where did Turing Award winners graduate?"

The anchor node is Turing Award.

Traversing Knowledge Graphs

- Answer path queries by traversing the KG.

“Where did Turing Award winners graduate?”

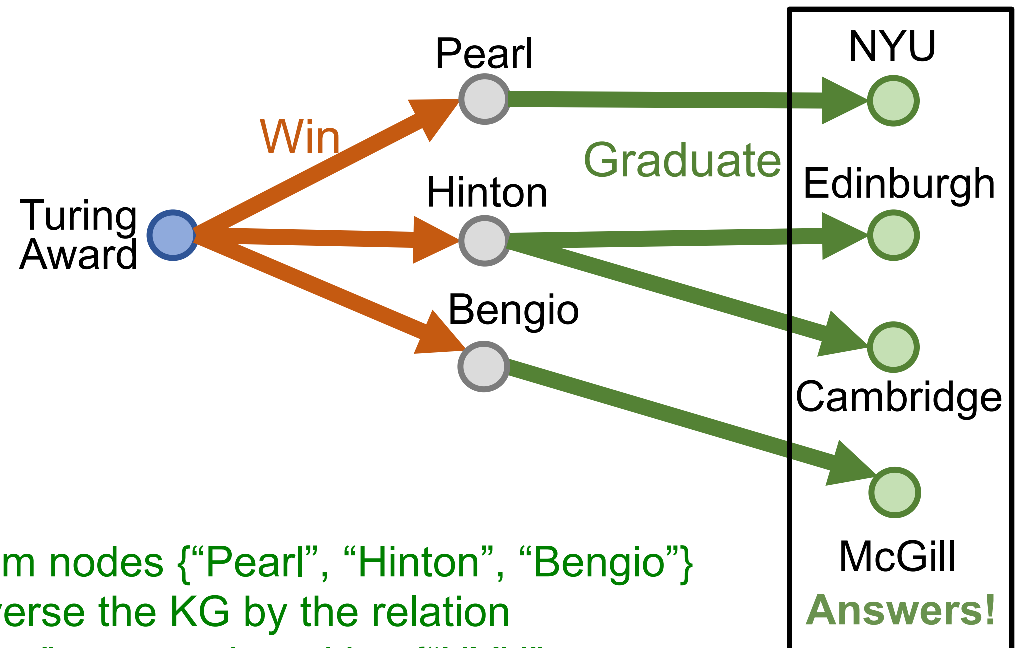


Start from the anchor node “Turing Award” and traverse the KG by the relation “Win”, we reach entities {“Pearl”, “Hinton”, “Bengio”}.

Traversing Knowledge Graphs

- Answer path queries by traversing the KG.

"Where did Turing Award winners graduate?"

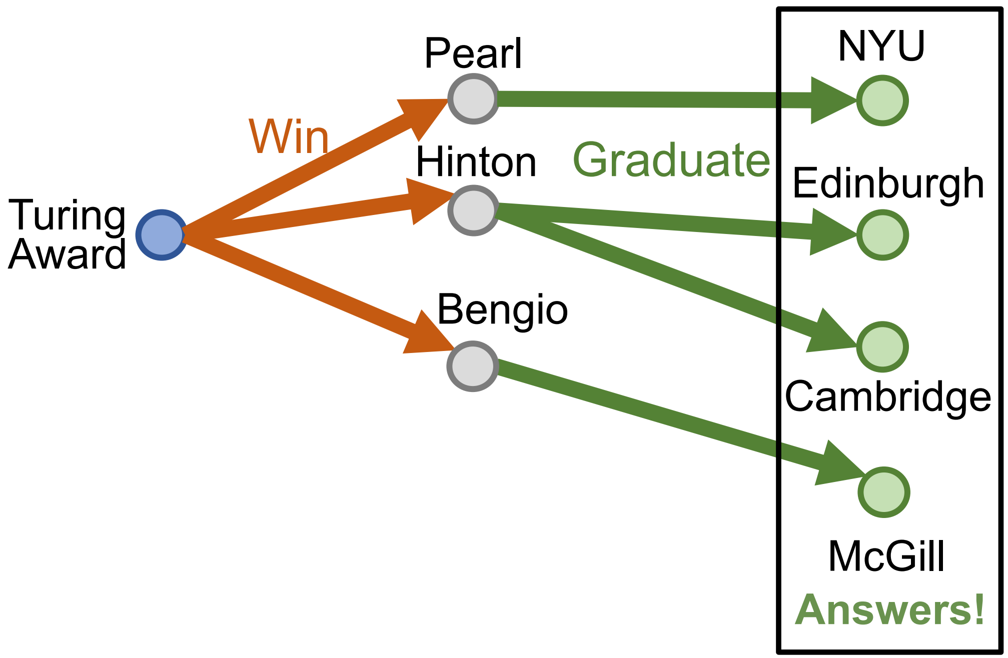


Start from nodes {"Pearl", "Hinton", "Bengio"} and traverse the KG by the relation "Graduate", we reach entities {"NYU", "Edinburgh", "Cambridge", "McGill"}. These are the answers to the query!

Traversing Knowledge Graphs

- Answer path queries by traversing the KG.

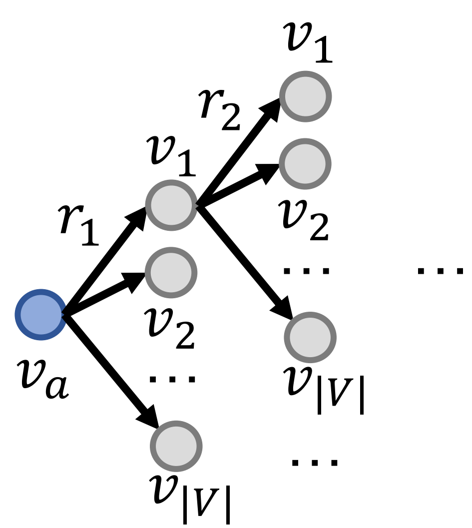
"Where did Turing Award winners graduate?"



What if KG is incomplete?

Answering Path Queries

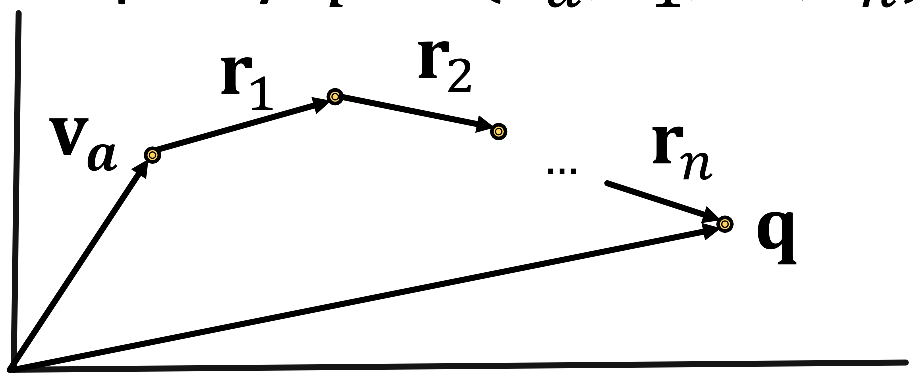
- Can we first do link prediction and then traverse the completed (probabilistic) KG?
- **No!** The completed KG is a **dense graph**!
- Time complexity of traversing a dense KG with $|V|$ entities to answer $(v_a, r_1, \ldots, r_n)$ of length n is $\mathcal{O}(|V|^n)$.



Traversing KG in Vector Space

- **Key idea: embed queries!**
  - Generalize TransE to multi-hop reasoning.

Given a path query $q = (v_a, r_1, \dots, r_n)$,



$$\mathbf{q} = \mathbf{v}_a + \mathbf{r}_1 + \cdots + \mathbf{r}_n$$

- Is v an answer to q?
  - Do a nearest neighbor search for all v based on $f_q(v) = ||\mathbf{q} - \mathbf{v}||$, time complexity is $\mathcal{O}(V)$.

Guu, Kelvin, John Miller, and Percy Liang. "Traversing knowledge graphs in vector space." arXiv preprint arXiv:1506.01094 (2015).

Traversing KG in Vector Space

- Embed path queries in vector space.

“Where did Turing Award winners graduate?”

Follow the computation graph:

Computation Graph

Embedding Space

Traversing KG in Vector Space

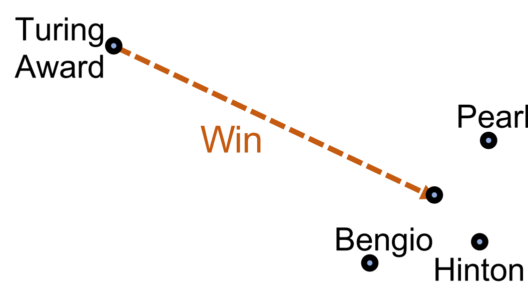
- Embed path queries in vector space.

“Where did Turing Award winners graduate?”

Follow the computation graph:

Computation Graph

Embedding Space

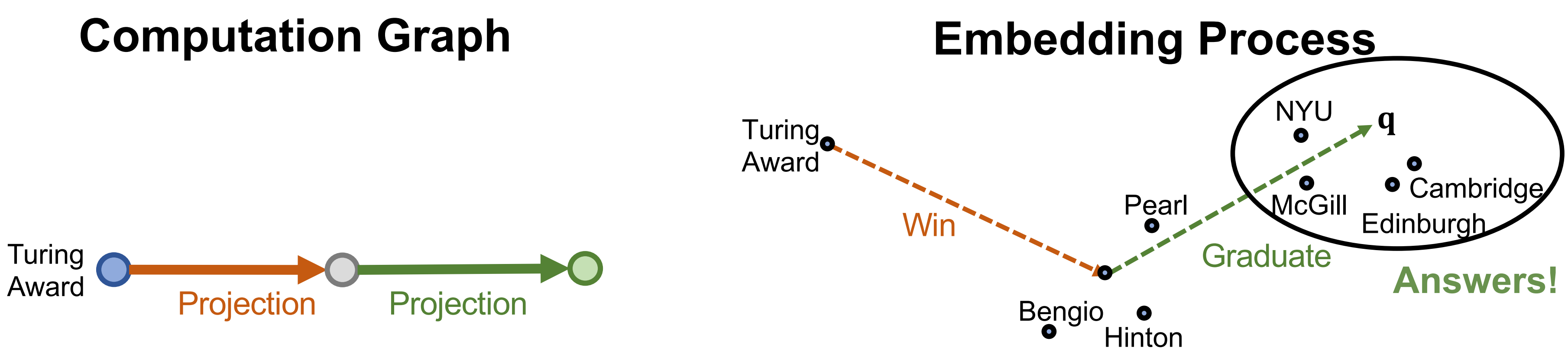


Traversing KG in Vector Space

- Embed path queries in vector space.

"Where did Turing Award winners graduate?"

Follow the computation graph:



Outline of Today's Lecture

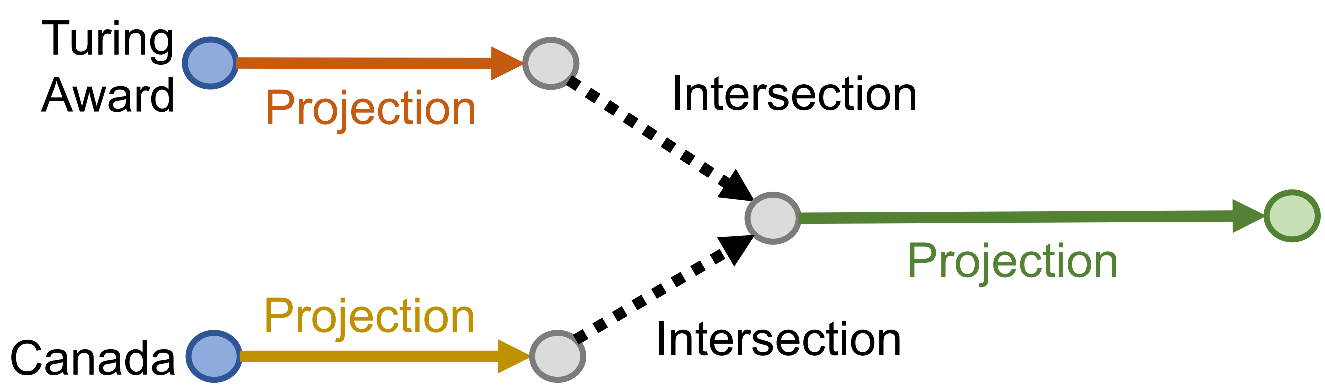
1. **Introduction to Knowledge Graphs**
2. **Link Prediction**
3. **Path Queries**
4. **Conjunctive Queries** 
5. **Query2Box: Reasoning with Box Embeddings**

Conjunctive Queries

- Can we answer more complex queries?
- What if we start from multiple anchor nodes?

"Where did Canadian citizens with Turing Award graduate?"

Computation graph of q:

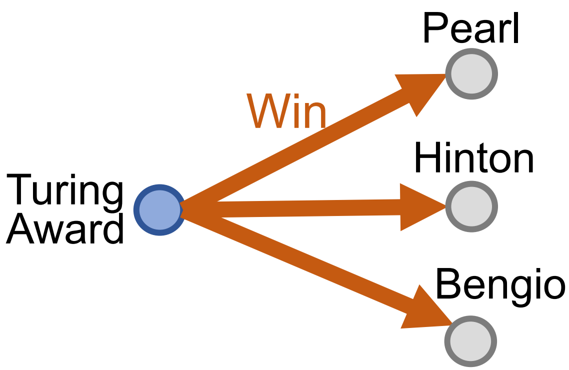


Conjunctive Queries

- Can we answer even more complex queries?

"Where did Canadian citizens with Turing Award graduate?"

Two anchor nodes: Canada and Turing Award.



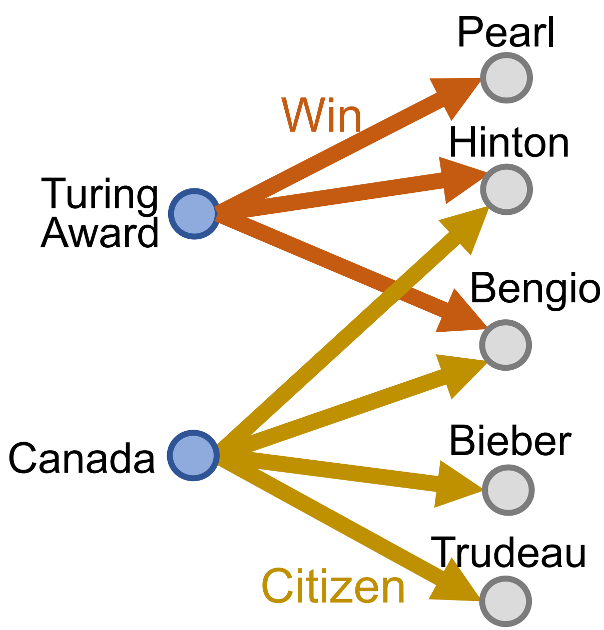
Start from the first anchor node "Turing Award", and traverse by relation "Win", we reach {"Pearl", "Hinton", "Bengio"} .

Conjunctive Queries

- Can we answer even more complex queries?

"Where did Canadian citizens with Turing Award graduate?"

Two anchor nodes: Canada and Turing Award.



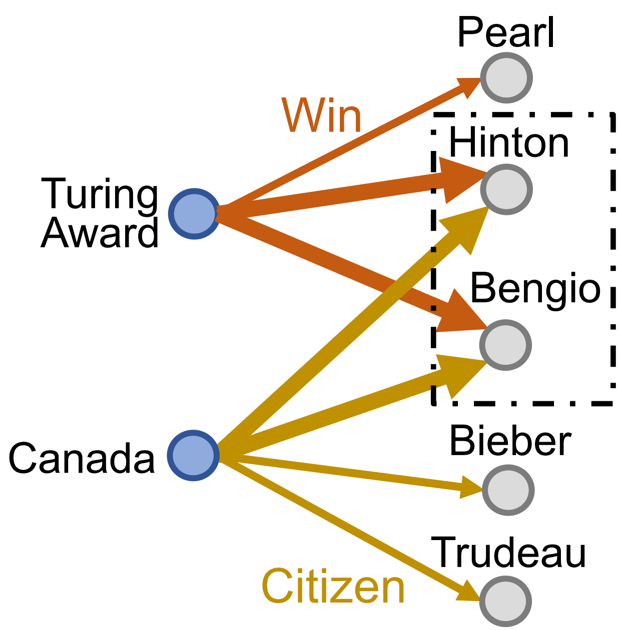
Start from the second anchor node "Canada", and traverse by relation "citizen", we reach { "Hinton", "Bengio", "Bieber", "Trudeau"}

Conjunctive Queries

- Can we answer even more complex queries?

"Where did Canadian citizens with Turing Award graduate?"

Two anchor nodes: Canada and Turing Award.



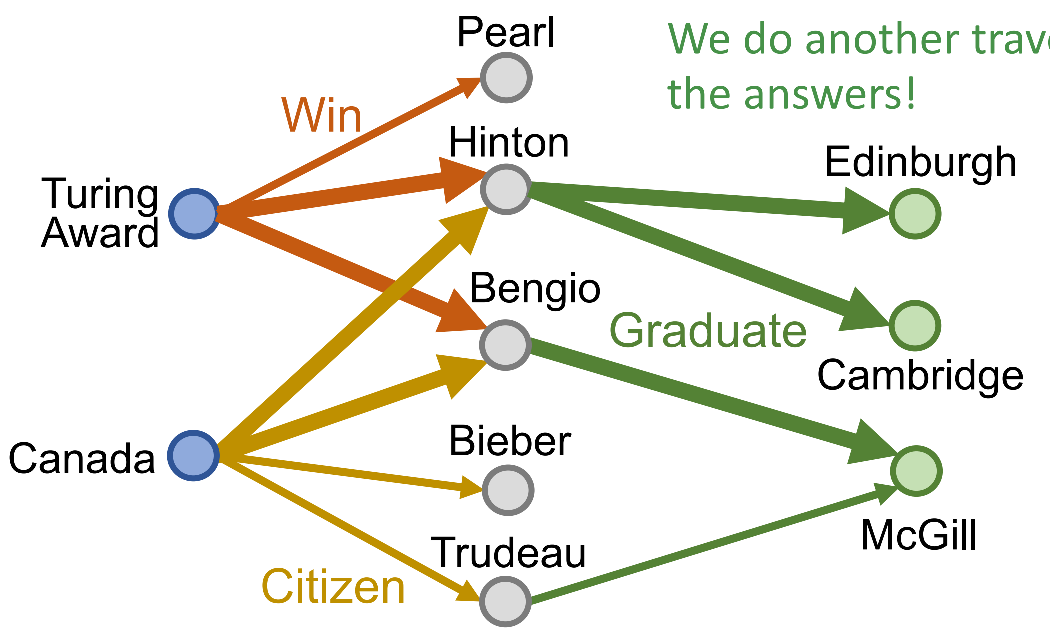
Then, we take intersection of the two sets and achieve {'Hinton', 'Bengio'}

Conjunctive Queries

- Can we answer even more complex queries?

"Where did Canadian citizens with Turing Award graduate?"

Two anchor nodes: Canada and Turing Award.



We do another traverse and arrive at the answers!

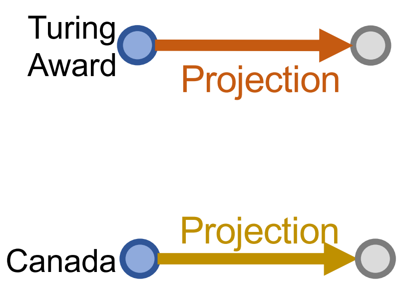
Traversing KG in Vector Space

- **Key Idea: embed queries in vector space**

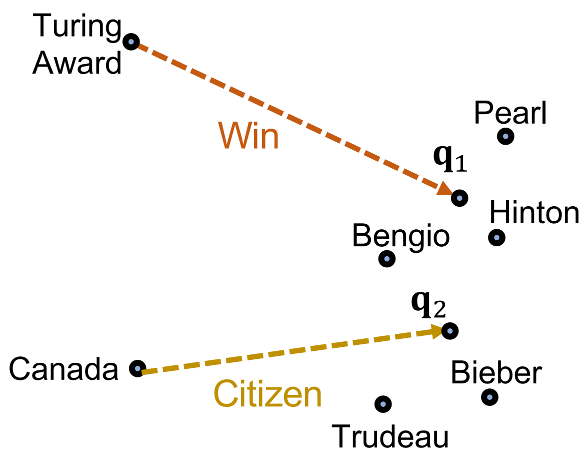
"Where did Canadian citizens with Turing Award graduate?"

Follow the computation graph:

Computation Graph



Embedding Space



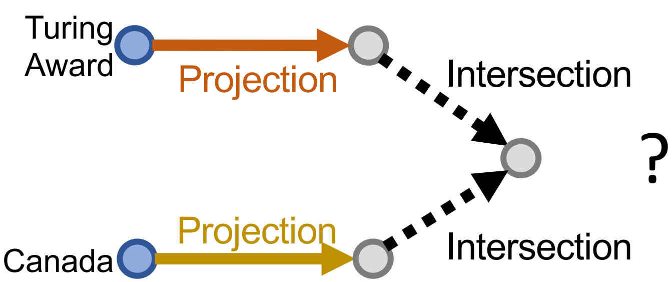
Traversing KG in Vector Space

- Key Idea: embed queries in vector space

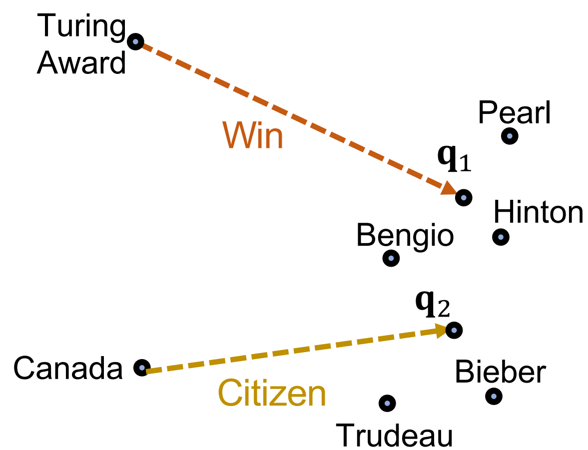
"Where did Canadian citizens with Turing Award graduate?"

Follow the computation graph:

Computation Graph



Embedding Process



Neural Intersection Operator

- How do we take intersection of several vectors in the embedding space?

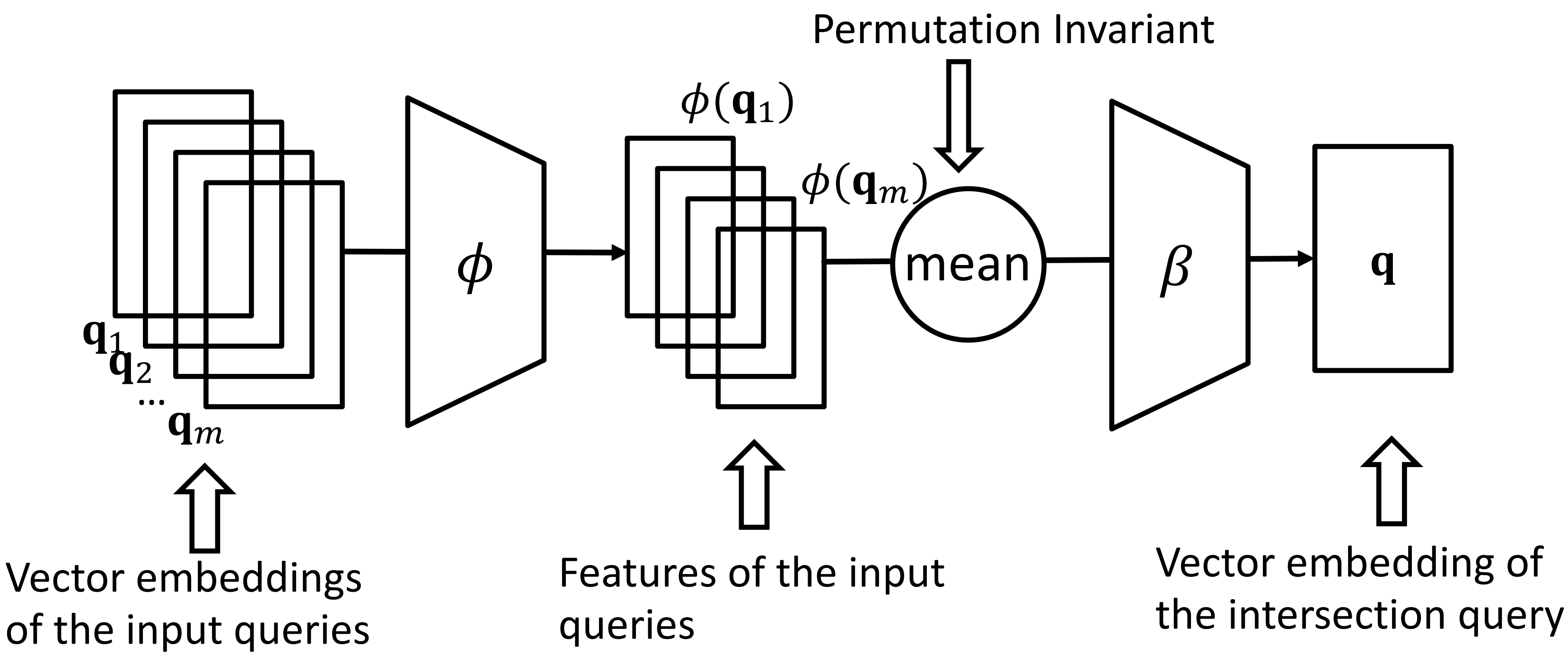
- Design a neural intersection operator $\mathcal{J}$
  - Input: current query embeddings $\mathbf{q}_1, \dots, \mathbf{q}_m$
  - Output: **intersection** query embedding $\mathbf{q}$
  - $\mathcal{J}$ should be **permutation invariant**:

$$\mathcal{J}(\mathbf{q}_1, \dots, \mathbf{q}_m) = \mathcal{J}(\mathbf{q}_{p(1)}, \dots, \mathbf{q}_{p(m)})$$

$[p(1), \dots, p(m)]$ is any permutation of $[1, \dots, m]$

Neural Intersection Operator

- DeepSets architecture



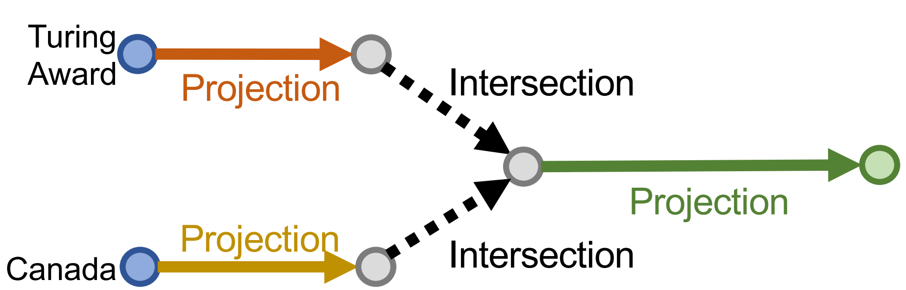
Traversing KG in Vector Space

- Key Idea: embed queries in vector space

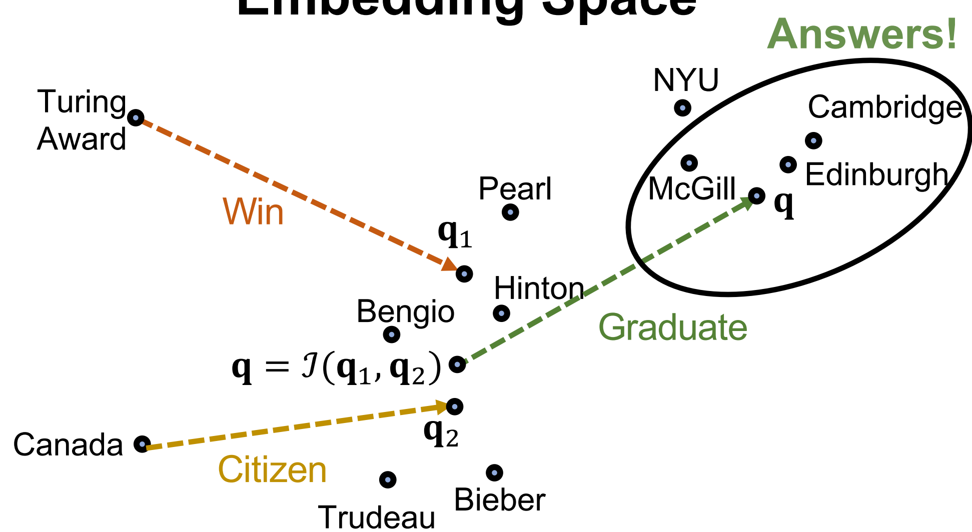
"Where did Canadian citizens with Turing Award graduate?"

Follow the computation graph:

Computation Graph



Embedding Space



Training

- Given an entity embedding $\mathbf{v}$ and a query embedding $\mathbf{q}$, the distance is $f_q(v) = ||\mathbf{q} - \mathbf{v}||$.

- Trainable parameters:
  - entity embeddings: $d|V|$
  - relation embeddings: $d|R|$
  - intersection operator ϕ, β: number of parameters does not depend on graph size

- **Same training strategy as TransE**

Whole Process

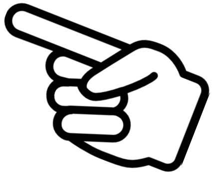
- **Training**:
  1. Sample a query q, answer v, negative sample v'.
  2. Embed the query $\mathbf{q}$.
  3. Calculate the distance $f_q(v)$ and $f_q(v')$.
  4. Optimize the loss $\mathcal{L}$.
- **Query evaluation**:
  1. Given a test query q, embed the query $\mathbf{q}$.
  2. For all v in KG, calculate $f_q(v)$.
  3. Sort the distance and rank all v.

Limitations

- Taking the intersection between two vectors is an operation that does **not follow intuition**.
- When we traverse the KG to achieve the answers, each step produces a set of reachable entities. How can we better model these sets?
- Can we define a **more expressive geometry** to embed the queries?

Outline

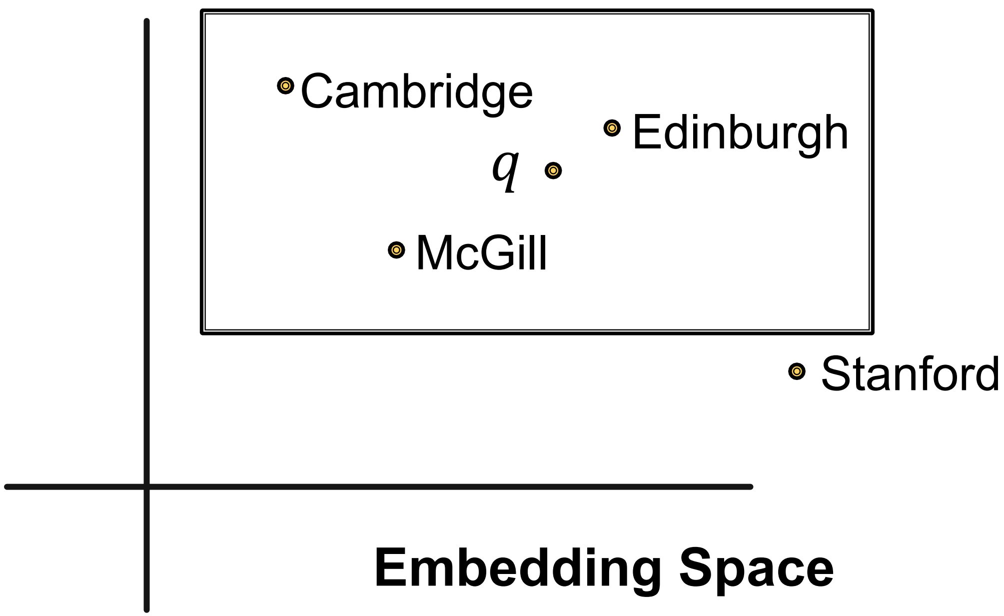
1. **Introduction to Knowledge Graphs**
2. **Knowledge Graph completion**
3. **Path Queries**
4. **Conjunctive Queries**
5. **Query2Box: Reasoning with Box Embeddings**



Box Embeddings

- Embed queries with hyper-rectangles (boxes)

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



Addressing Limitations

- Taking intersection between two vectors is an operation that does not follow intuition.
  - Intersection of boxes is well-defined!
- When we traverse the KG to achieve the answers, each step produces a set of reachable entities. How can we better model these sets?
  - Boxes are a **powerful abstraction**, as we can project the center and control the offset to model the set of entities enclosed in the box.

Embed with Box Embeddings

- Parameters:
  - entity embeddings: $d|V|$
    - entities are seen as zero-volume boxes
  - relation embeddings: $2d|R|$
    - augment each relation with an offset
  - intersection operator ϕ, β: number of parameters does not depend on graph size
    - New operator, inputs are boxes and output is a box

Embed with Box Embedding

- Embed queries in vector space

"Where did Canadian citizens with Turing Award graduate?"

Note that computation graph stays the same!

Follow the computation graph:

Computation Graph

Embedding Space

Embed with Box Embedding

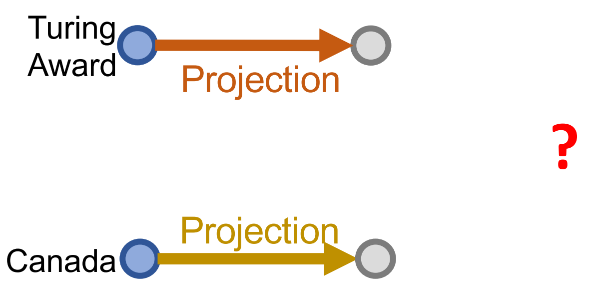
- Embed queries in vector space

"Where did Canadian citizens with Turing Award graduate?"

Note that computation graph stays the same!

Follow the computation graph:

Computation Graph

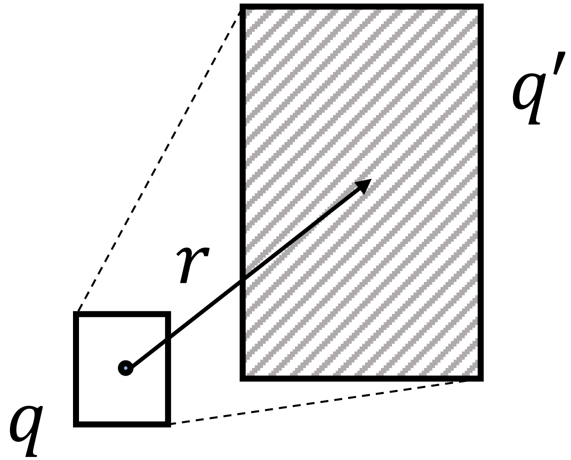


Embedding Process

Projection Operator

- Geometric Projection Operator $\mathcal{P}$
- $\mathcal{P}$: Box × Relation → Box

$$Cen(q') = Cen(q) + Cen(r)$$
$$Off(q') = Off(q) + Off(r)$$



Embed with Box Embedding

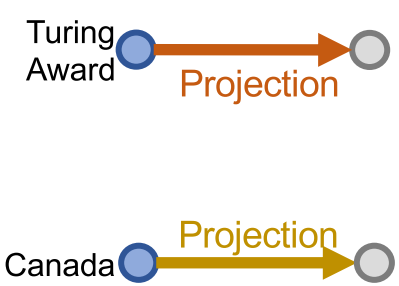
- Embed queries in vector space

“Where did Canadian citizens with Turing Award graduate?”

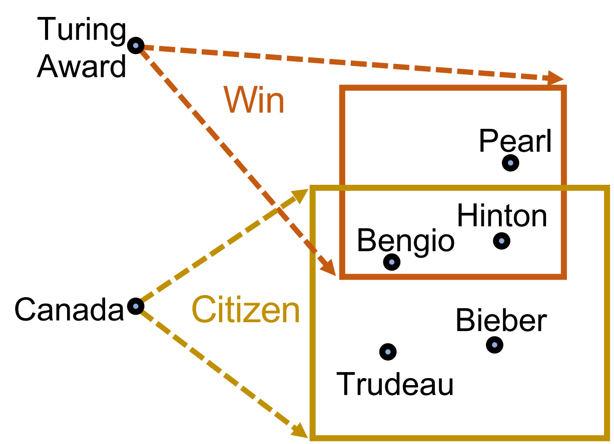
Note that computation graph stays the same!

Follow the computation graph:

Computation Graph



Embedding Space



Embed with Box Embedding

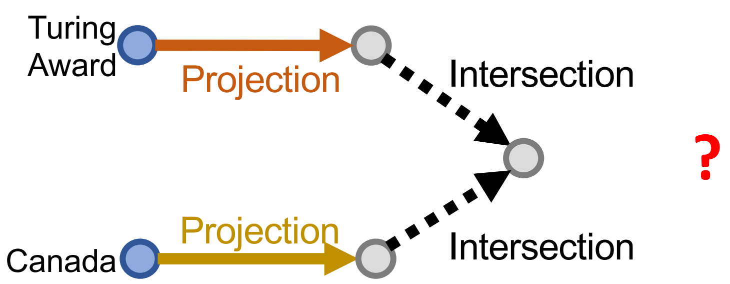
- Embed queries in vector space

"Where did Canadian citizens with Turing Award graduate?"

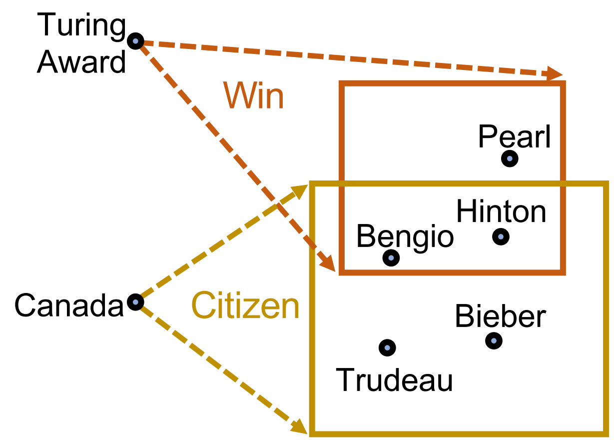
Note that computation graph stays the same!

Follow the computation graph:

Computation Graph

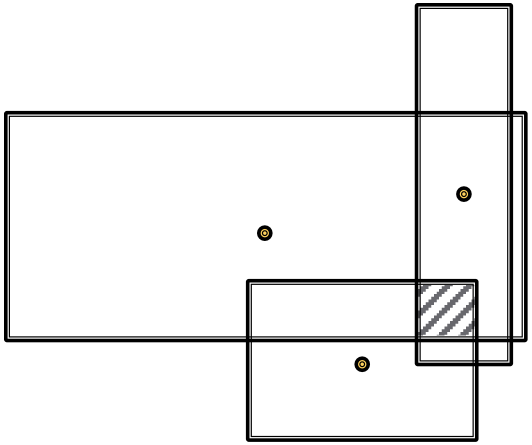


Embedding Space



Intersection Operator

- Geometric Intersection Operator $\mathcal{J}$
- $\mathcal{J}$: Box $\times \cdots \times$ Box $\rightarrow$ Box
  - The new center is a weighted average.
  - The new offset shrinks.



Intersection Operator

- Geometric Intersection Operator $\mathcal{J}$
- $\mathcal{J}$: Box $\times \cdots \times$ Box $\rightarrow$ Box

dimension-wise product

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

weight

$$Off(q_{inter})$$
$$= \min(Off(q_1), \ldots, Off(q_n))$$
$$\odot\, \sigma(Deepsets(\mathbf{q}_1, \ldots, \mathbf{q}_n))$$

guarantees shrinking

Sigmoid function:
squashes output in (0,1)

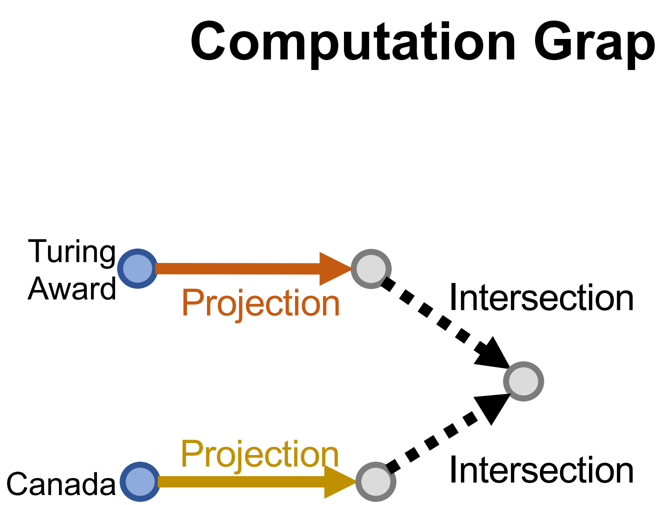
Embed with Box Embedding

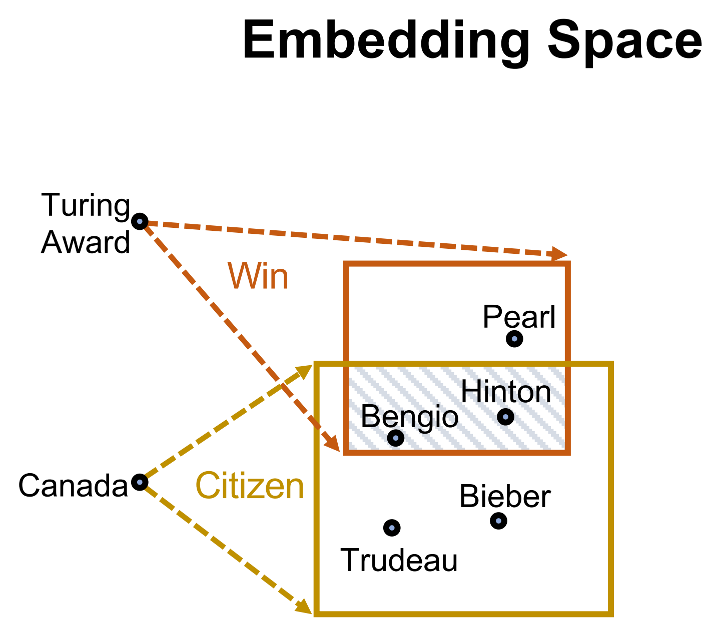
- Embed queries in vector space

"Where did Canadian citizens with Turing Award graduate?"

Note that computation graph stays the same!

Follow the computation graph:





Embed with Box Embedding

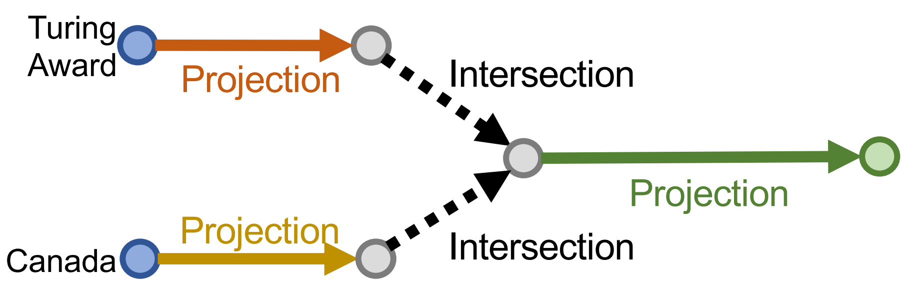
- Embed queries in vector space

"Where did Canadian citizens with Turing Award graduate?"

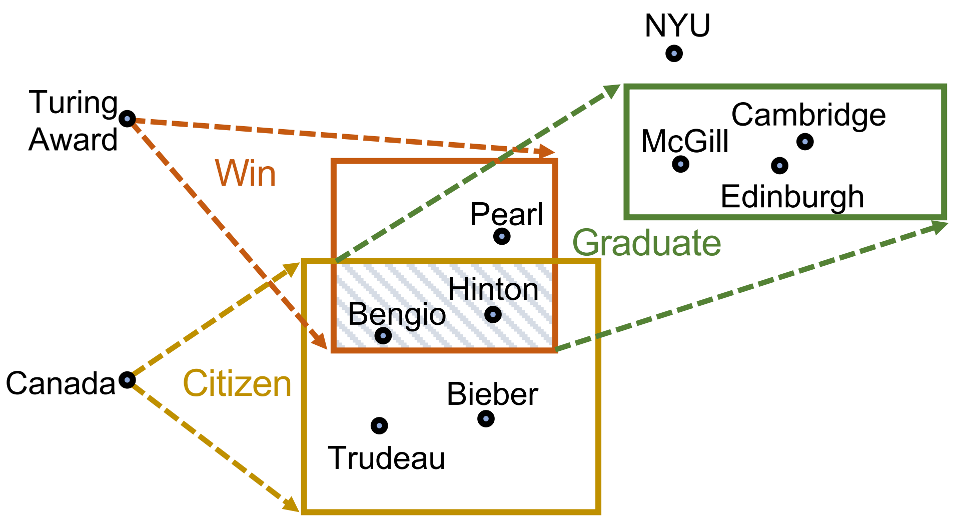
Note that computation graph stays the same!

Follow the computation graph:

Computation Graph



Embedding Space

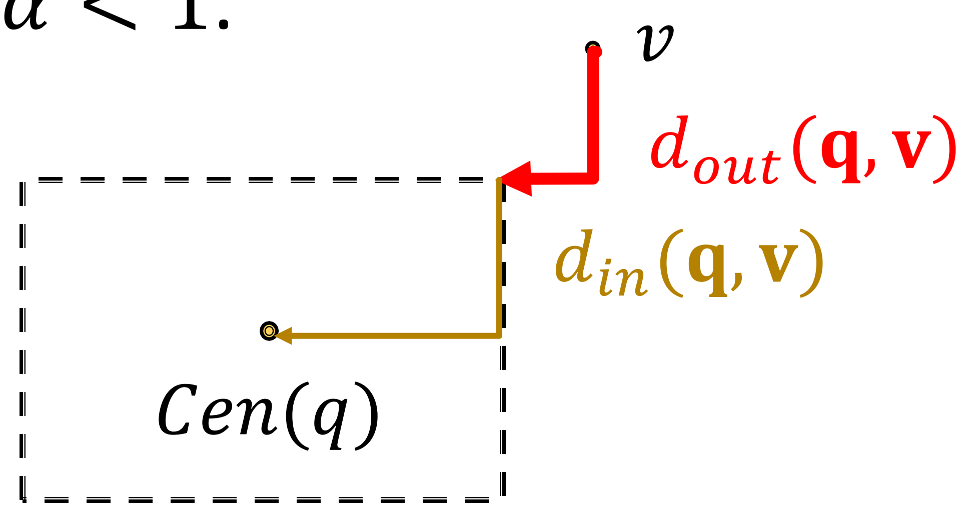


Entity-to-Box Distance

- Given a query box $\mathbf{q}$ and entity vector $\mathbf{v}$,

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

where $0 < \alpha < 1$.

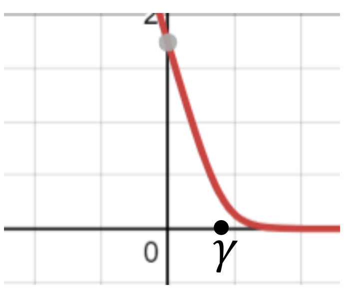


Training Query2box

- Given a set of queries and answers,

$$\mathcal{L} = -\log \sigma\big(\gamma - d_{box}(q, v)\big) - \log \sigma\big(d_{box}(q, v_i') - \gamma\big)$$

$-\log \sigma\big(\gamma - d_{box}(q, v)\big)$
minimize loss → minimize $d_{box}(q, v)$



$-\log \sigma\big(d_{box}(q, v') - \gamma\big)$
minimize loss → maximize $d_{box}(q, v')$

Relation Patterns

- Can query2box handle different relation patterns?

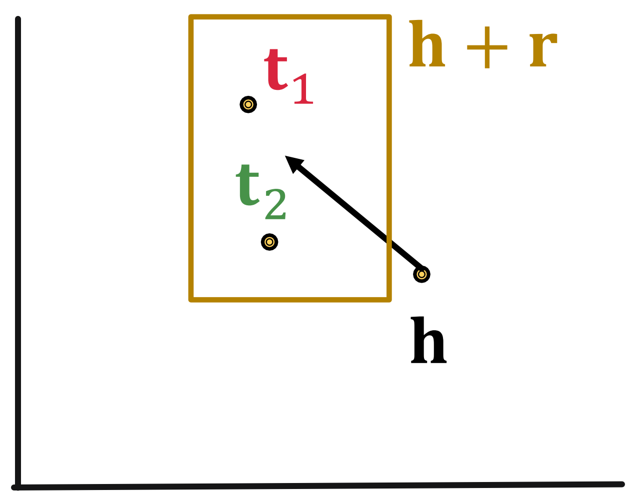
Embedding	Symmetry	Composition	One-to-many
TransE	✗	✓	✗
TransH	✓	✗	✓
Query2Box	✓	✓	✓

For details please check the paper https://openreview.net/forum?id=BJgr4kSFDS

N-ary Relations in query2box

- 1-to-N, N-to-1, N-to-N relations.
- **Example**: Both (h, r, t_1) and (h, r, t_2) exist.

- Box Embedding can handle since t_1 and t_2 will be mapped to different locations in the box of (h, r). ✓



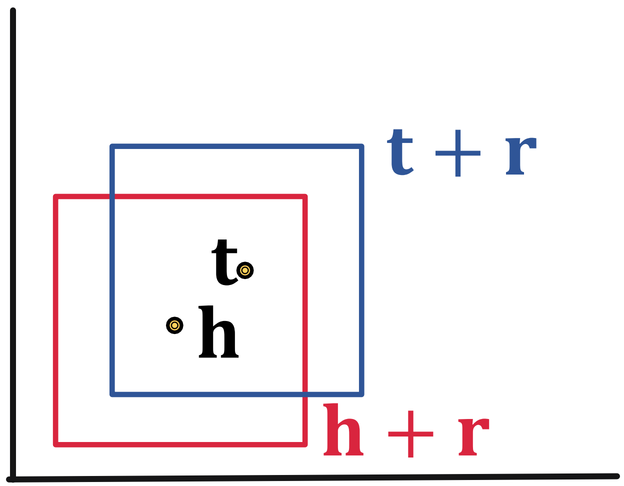
Symmetric Relations in query2box

- Symmetric Relations:

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

- **Example**: Family, Roommate
- Box Embedding

$$Cen(r) = 0 \checkmark$$

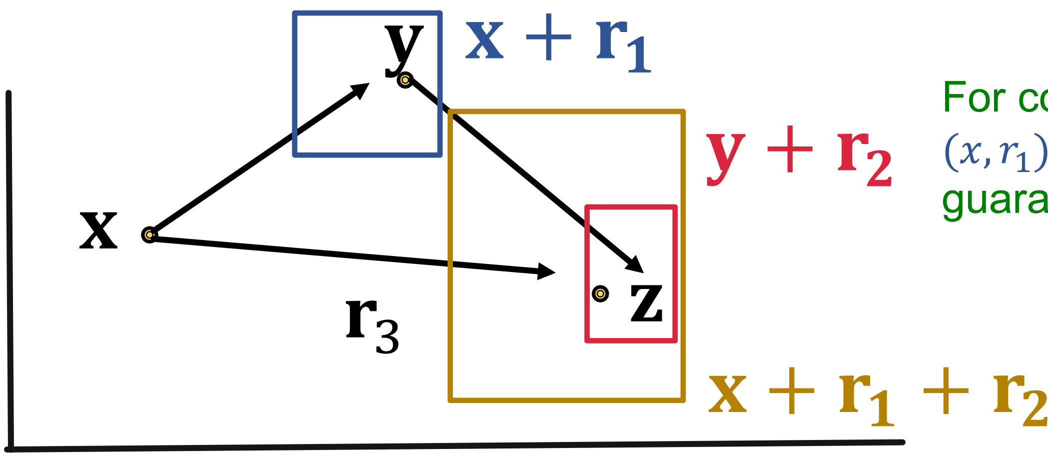


For symmetric relations r, we could assign $Cen(r) = 0$. In this case, as long as t is in the box of (h,r), it is guaranteed that h is in the box of (t,r). So we have $r(h,t) \Rightarrow r(t,h)$

Composition Relations in query2box

- Composition Relations:
  $$r_1(x,y) \wedge r_2(y,z) \Rightarrow r_3(x,z) \quad \forall x,y,z$$
- **Example**: My mother's husband is my father.
- Box Embedding

$$\mathbf{r}_3 = \mathbf{r}_1 + \mathbf{r}_2 \checkmark$$



For composition relations, if y is in the box of (x, r_1) and z is in the box of (y, r_2), it is guaranteed that z is in the box of $(x, r_1 + r_2)$.

EPFO queries

- Can we embed even more complex queries?

"Where did Canadians with Turing Award or Nobel graduate?"

- **Conjunctive queries + disjunction** is called Existential Positive First-order (**EPFO**) queries.

- Can we also design a disjunction operator and embed EPFO queries in low-dimensional vector space? **YES!**

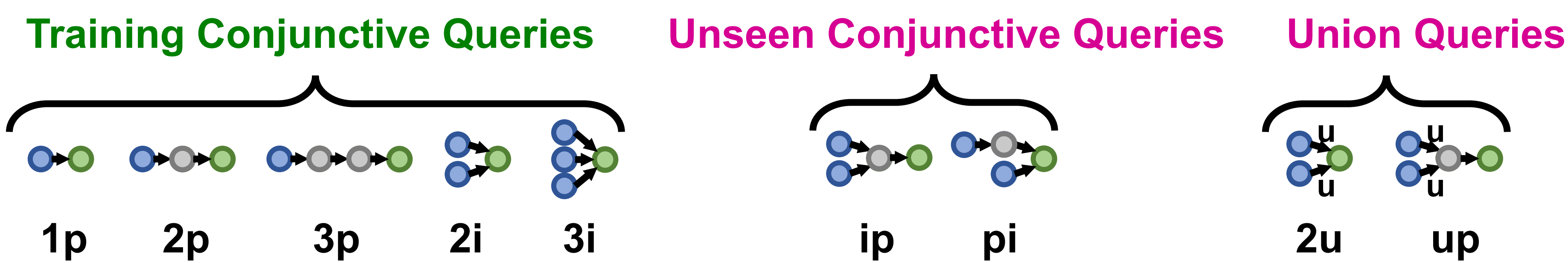
For details please check the paper https://openreview.net/forum?id=BJgr4kSFDS

Experiments

- Datasets: FB15K, FB15K-237

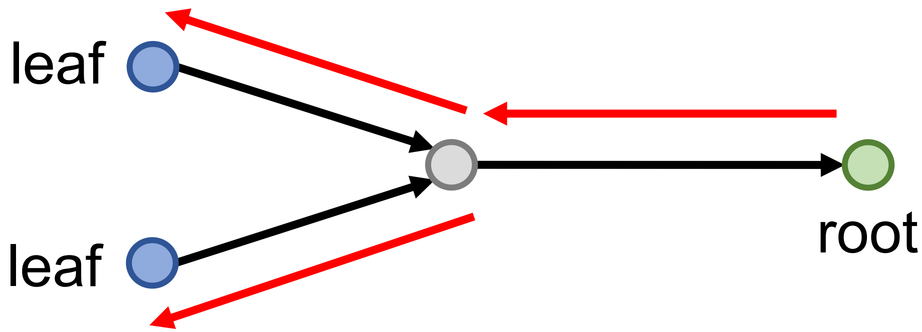
Dataset	Entities	Relations	Training Edges	Validation Edges	Test Edges	Total Edges
FB15k	14,951	1,345	483,142	50,000	59,071	592,213
FB15k-237	14,505	237	272,115	17,526	20,438	310,079

- Goal: can the model discover true answers that cannot be achieved by traversing the KG?
  - Training KG: Training Edges
  - Validation KG: Training Edges + Validation Edges
  - Test KG: Training Edges + Validation Edges + Test Edges
- Queries:

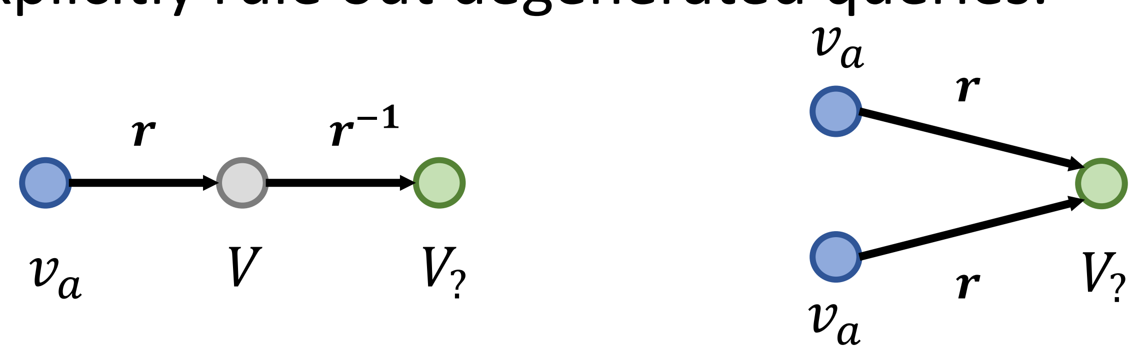


Query Generation

- Given a query structure, use pre-order traversal (traverse from root to leaves) to assign an entity/relation for every node/edge.

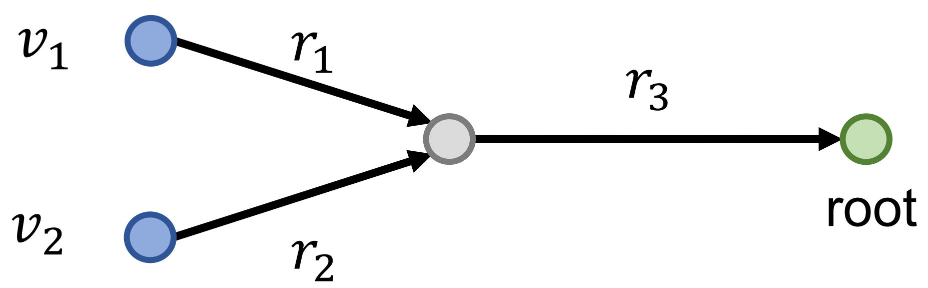


- We explicitly rule out degenerated queries.



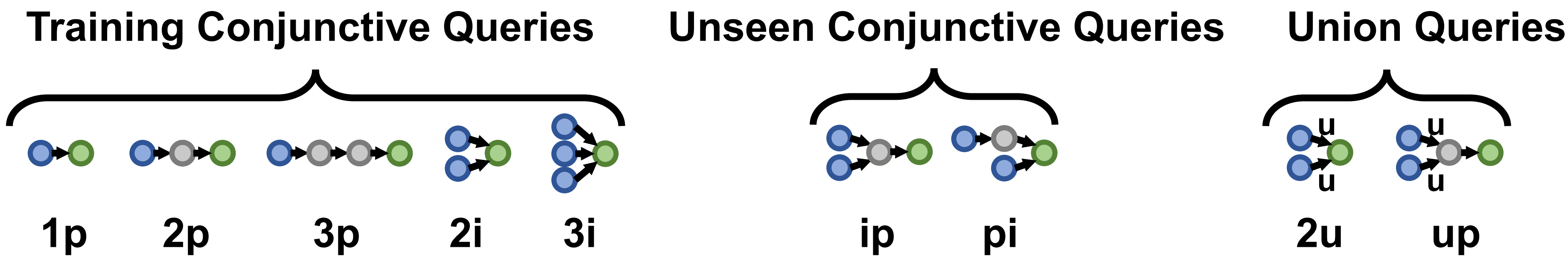
Query Generation

- After instantiation, run post-order traversal (traverse from leaves v_1, v_2 to root) to achieve all answers.



- For test queries, we guarantee that they cannot be fully answered on training/validation KG.

Query Statistics



Queries	Training		Validation		Test	
Dataset	1p	others	1p	others	1p	others
FB15k	273,710	273,710	59,097	8,000	67,016	8,000
FB15k-237	149,689	149,689	20,101	5,000	22,812	5,000

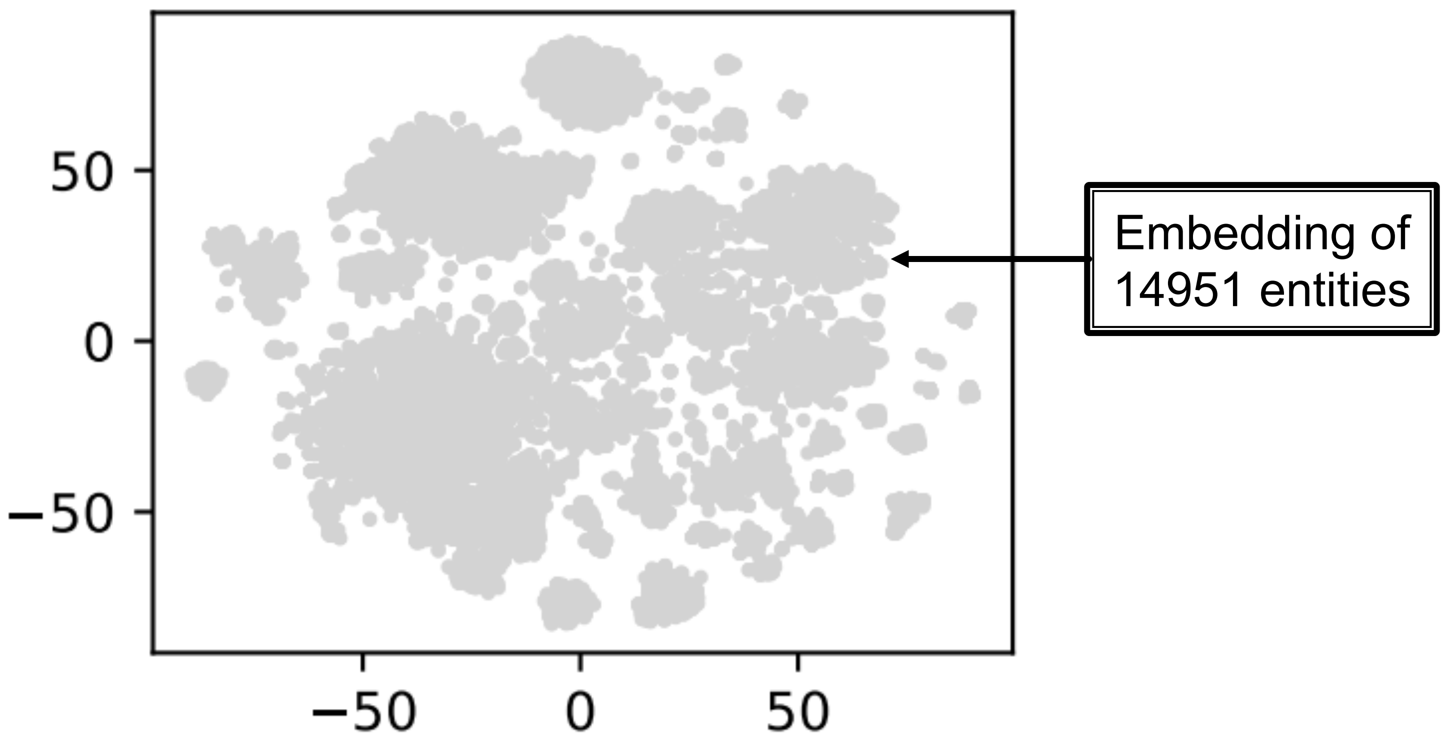
Visualization

- What does query2box actually learn?

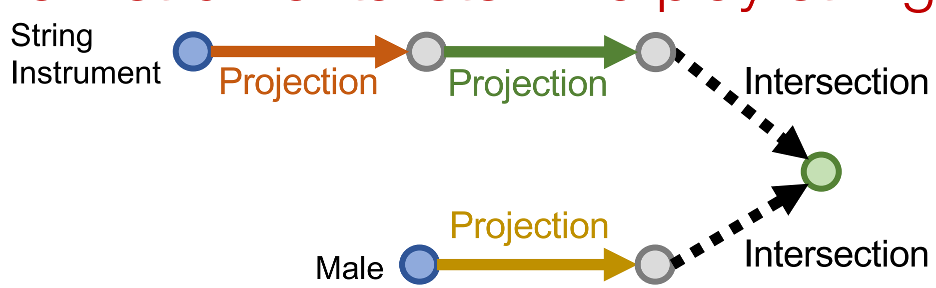
Example: "*List male instrumentalists who play string instruments*"

- We use T-SNE to reduce the embedding space to a 2-dimensional space, in order to **visualize the query results**

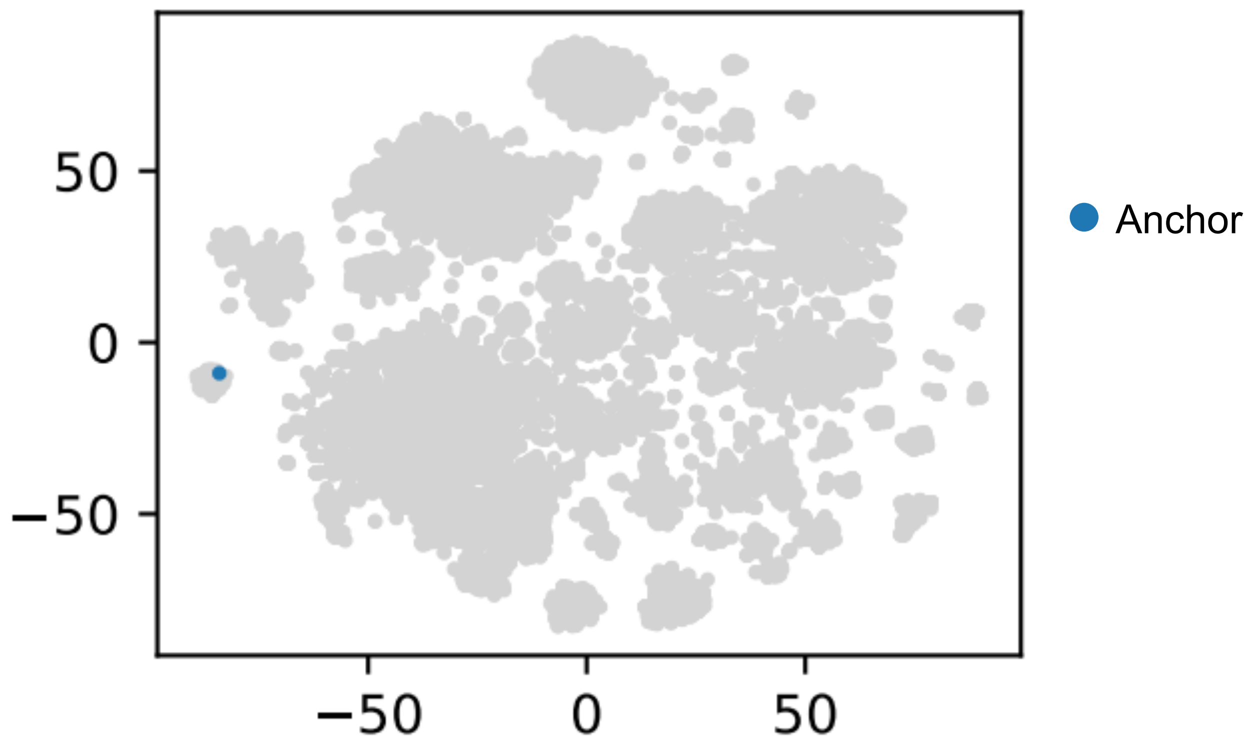
Embedding Space



"List male instrumentalists who play string instruments"

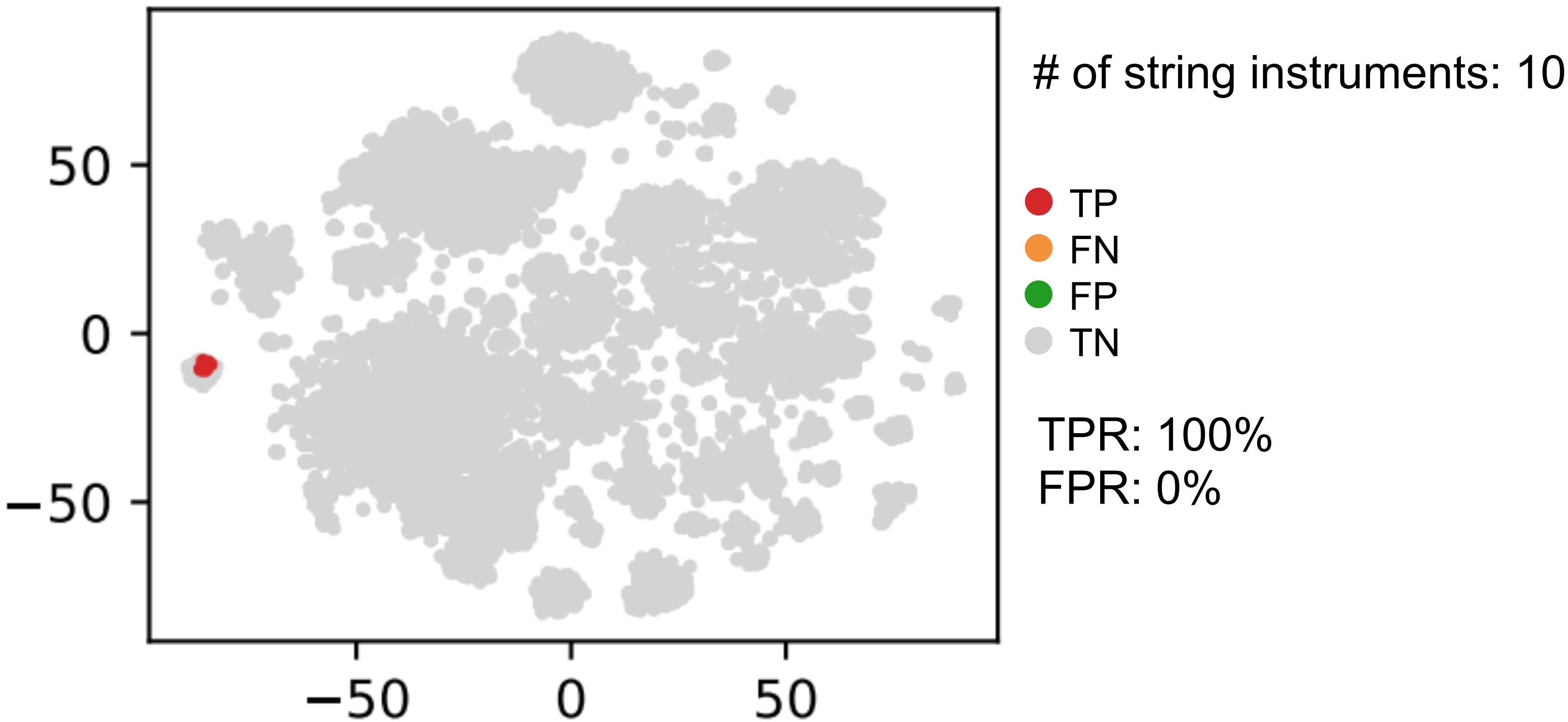


Embedding Space



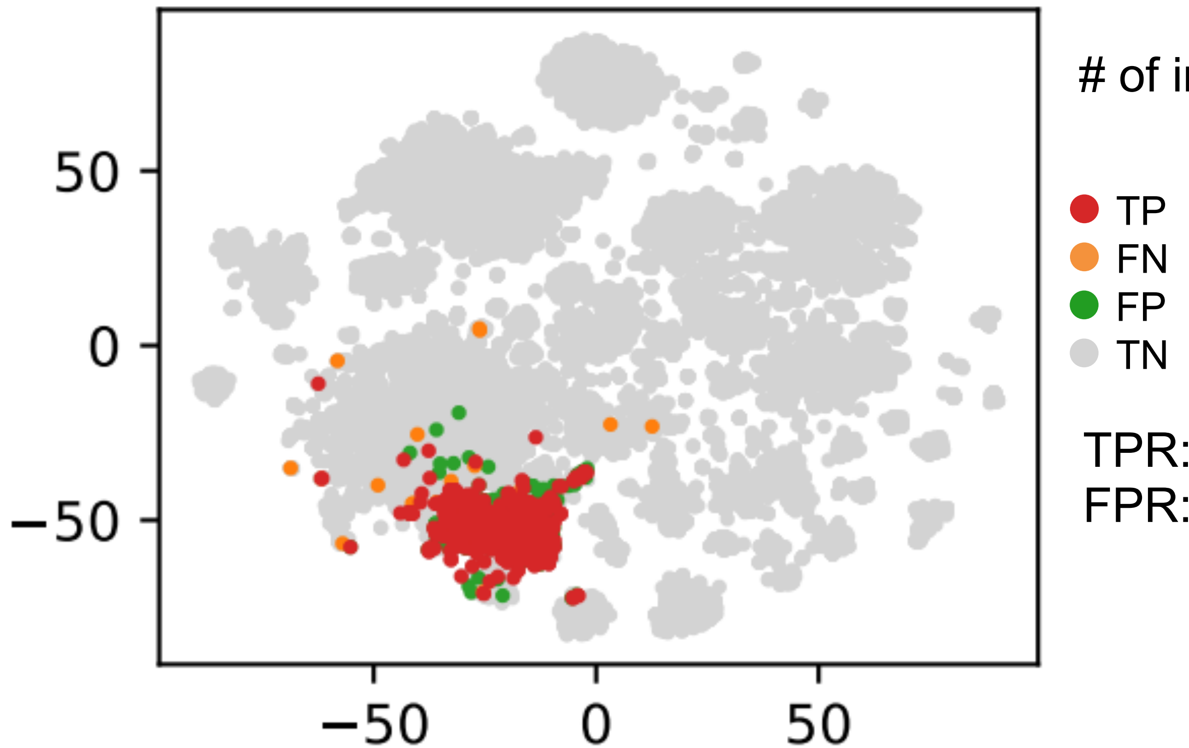
“List male instrumentalists who play string instruments”

Embedding Space



"List male instrumentalists who play <u>string instruments</u>"

Embedding Space

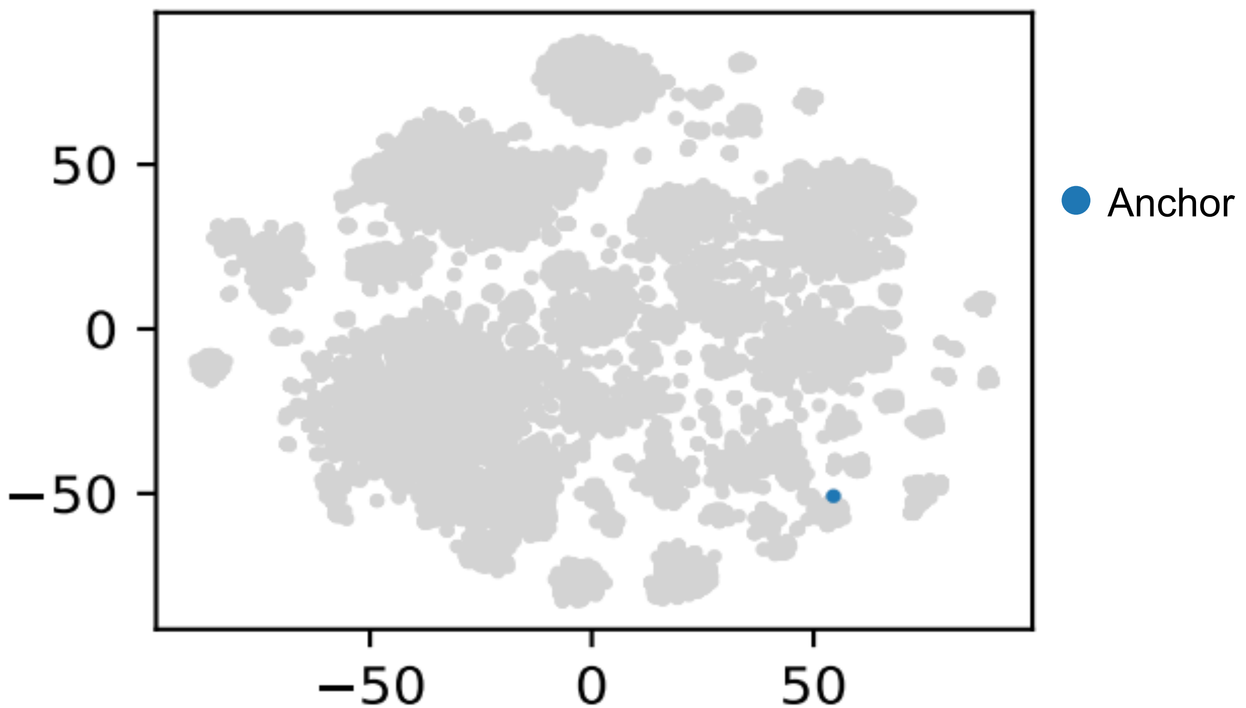


# of instrumentalists: 472

TPR: 98.4%
FPR: 0.01%

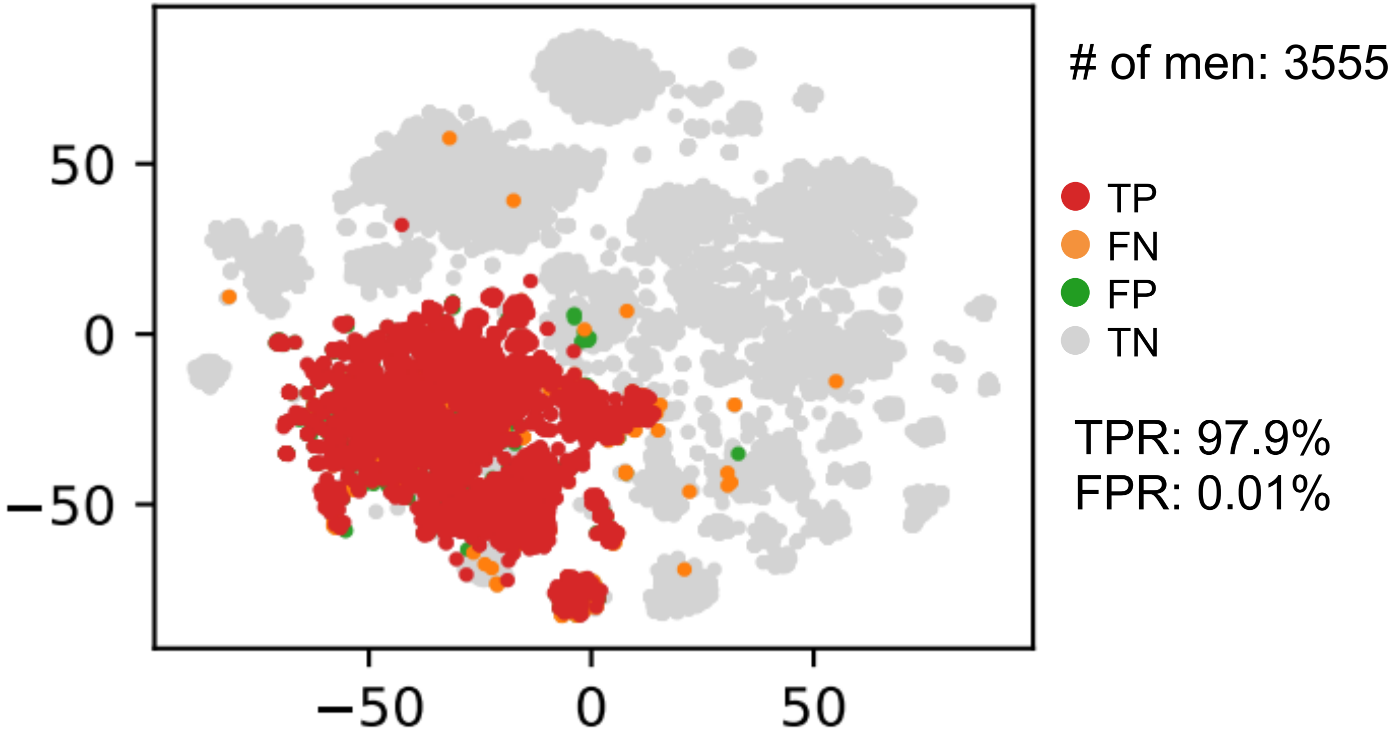
"List male instrumentalists who play string instruments"

Embedding Space



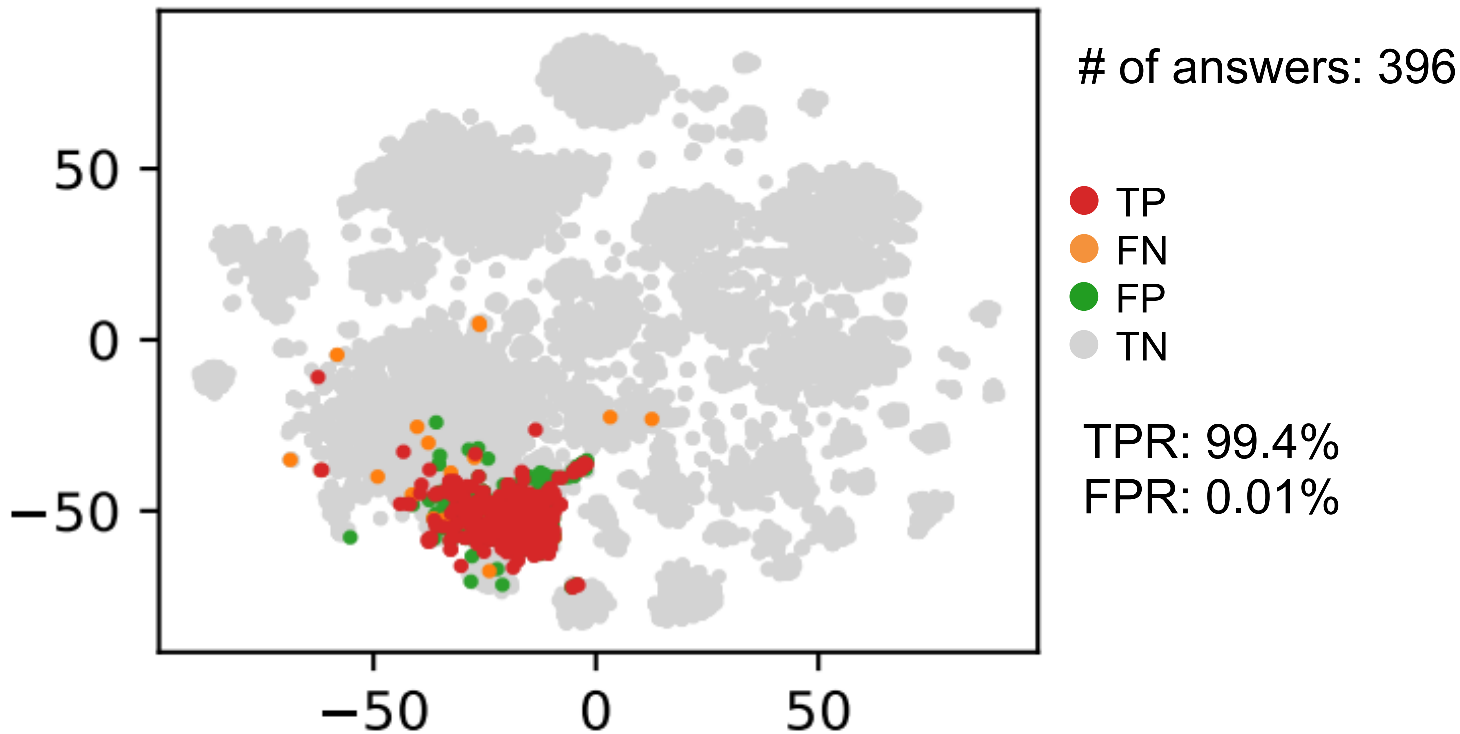
“List male instrumentalists who play string instruments”

Embedding Space



"List male instrumentalists who play string instruments"

Embedding Space



"List male instrumentalists who play string instruments"