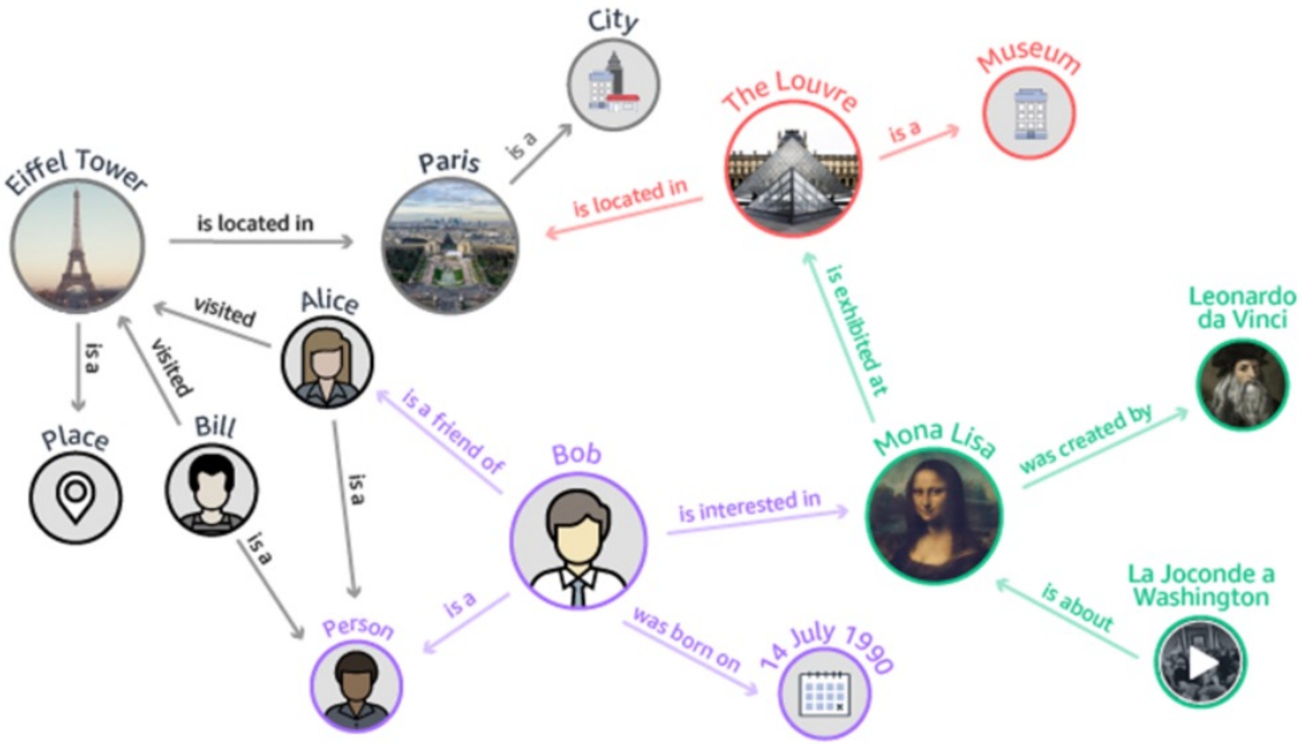
Complex Logical Reasoning over Knowledge Graphs

Bo Xiong

07.01.2022

Knowledge Graphs

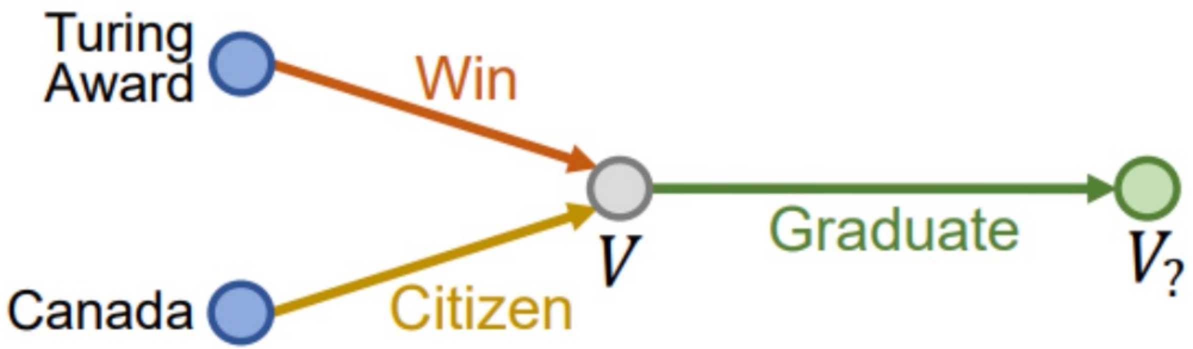
- Knowledge Graphs (KGs) are **heterogenous** graphs
- Facts are represented as triples *(h, r, t)*



(Complex) Logical Reasoning

- Reasoning over **complex logical query**
- For example:
  - Where did Canadian Turing Award winners graduate?

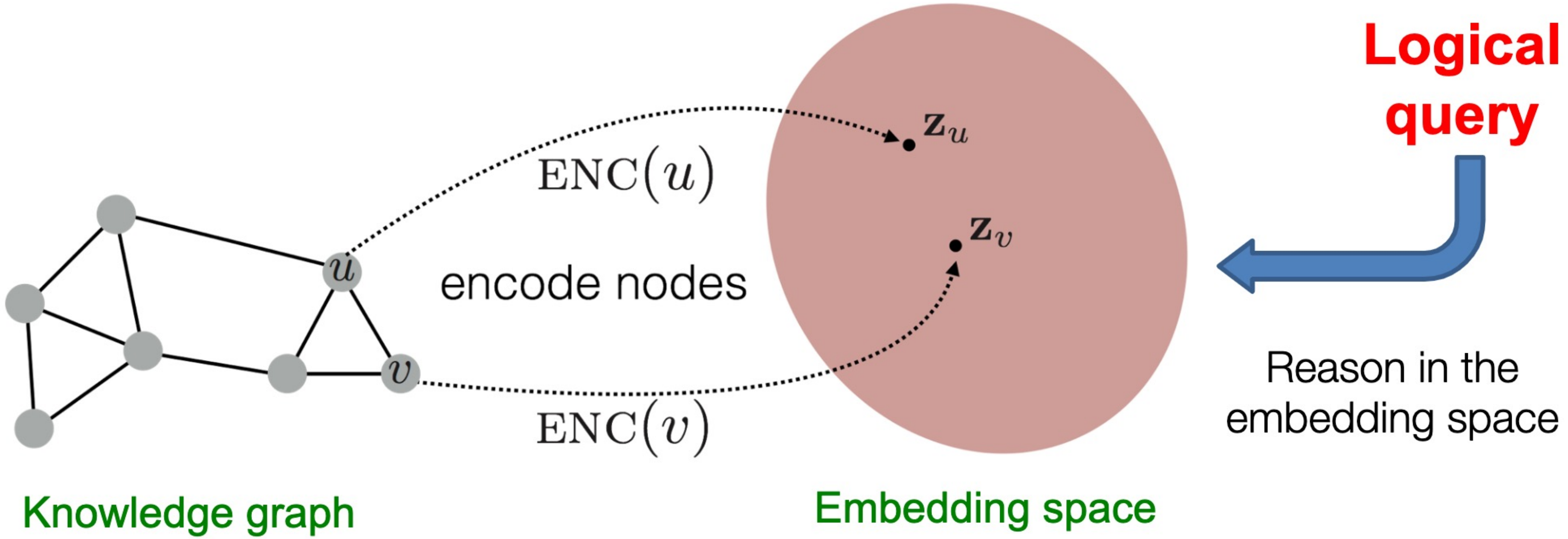
$$q = V_? \,.\, \exists V : Win(TuringAward, V) \wedge Citizen(Canada, V) \wedge Graduate(V, V_?)$$



Challenges

- Noise and incompleteness
- Lack of schema, or quite large schema (65k for DBpedia)
- Massive Size
  - Google knowledge graph: 570 million entities and 18 billion facts
  - Yago: 10 million entities and 120 million facts
- Slow query (e.g., subgraph matching)

Embedding based reasoning



- Map queries into embedding space.
- Learn to reason in that space

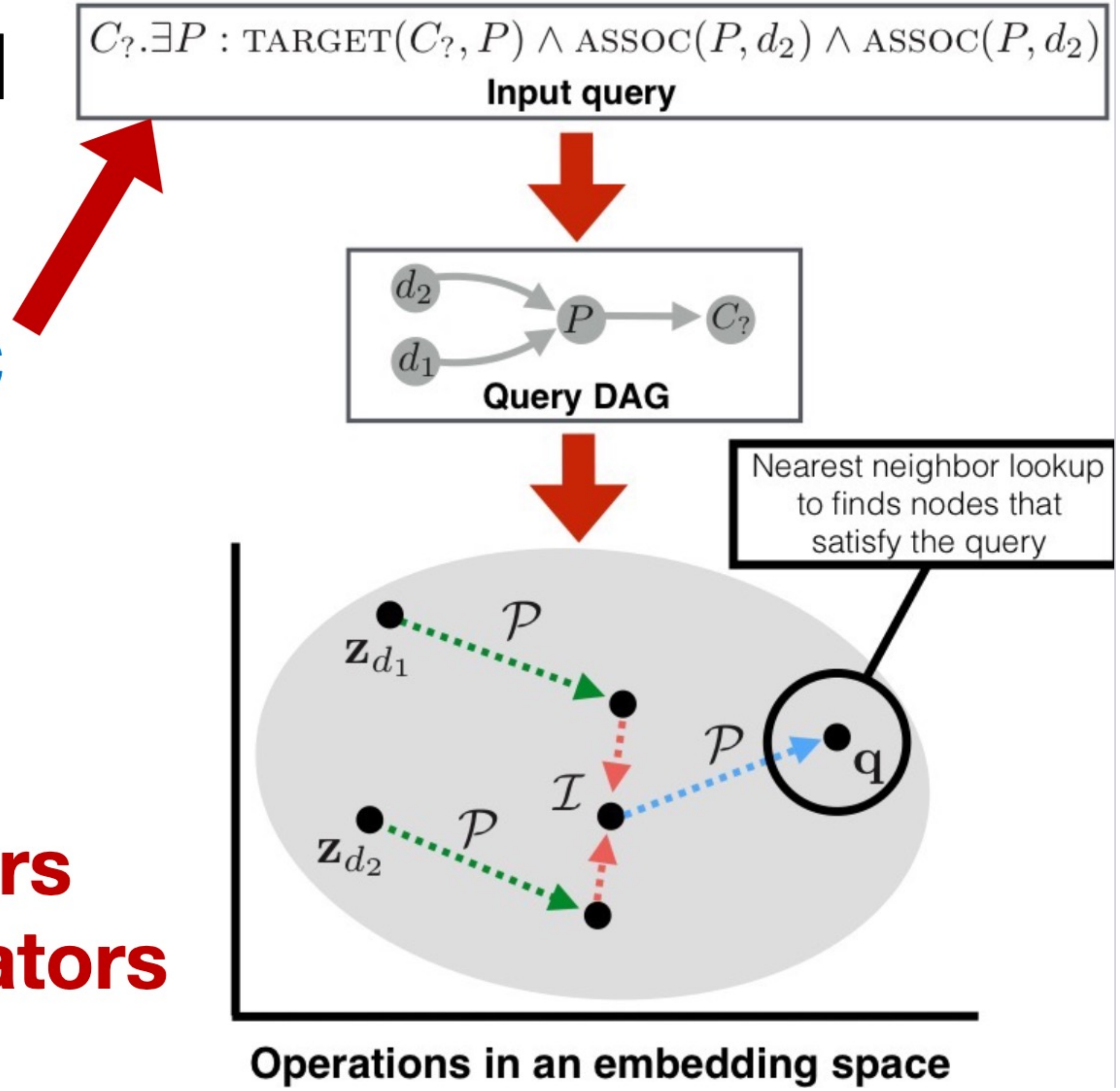
[Embedding Logical Queries on Knowledge Graphs.] Hamilton, et al., NeurIPS 2018]
[Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings.] Ren, et al., ICLR 2020]

General Framework

Goal: Answer logical queries

E.g.: "**Predict drugs C** likely *target* proteins X *associated* with diseases d_1 and d_2"

Idea: Logical operators become spatial operators



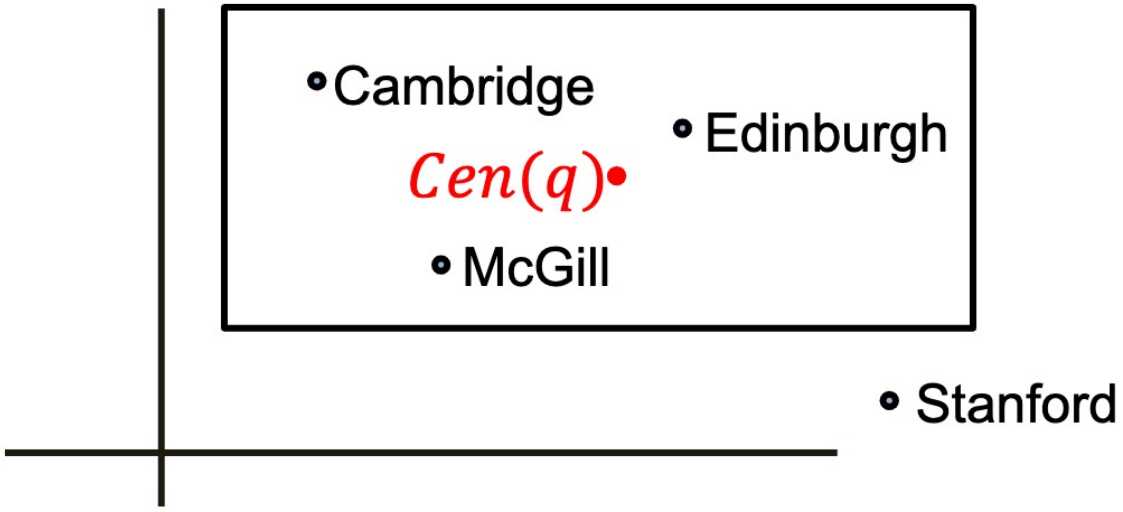
How to embed query?

- Conjunctive Query
  - **Query2box:** Reasoning over Knowledge Graphs in Vector Space using Box Embeddings, https://arxiv.org/abs/2002.05969
- First-Order Logic Query
  - **Beta Embeddings** for Multi-Hop Logical Reasoning in Knowledge Graphs,https://arxiv.org/abs/2010.11465
  - **ConE:** Cone Embeddings for Multi-Hop Reasoning over Knowledge Graphs, https://arxiv.org/abs/2110.13715
- …

Query2Box

Query2Box embedding:

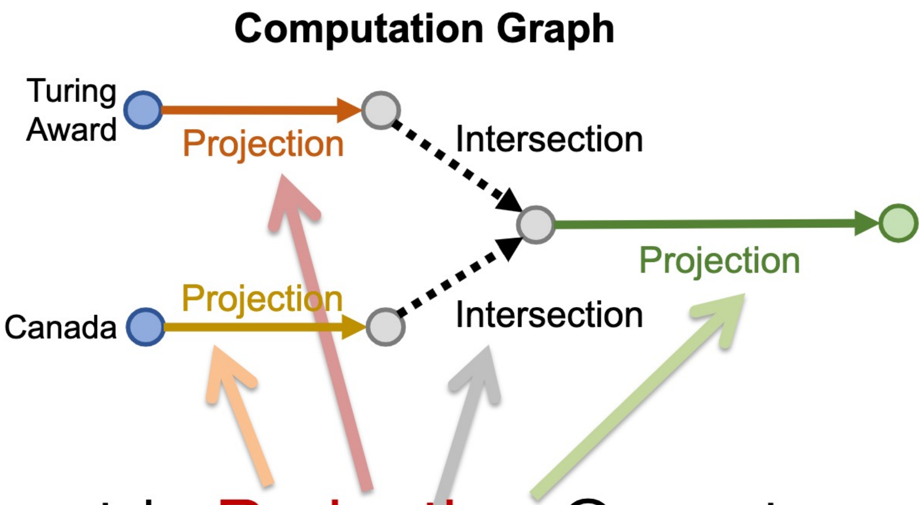
Embed queries with hyper-rectangles (boxes): $\mathbf{q} = (Cen(q), Off(q))$.



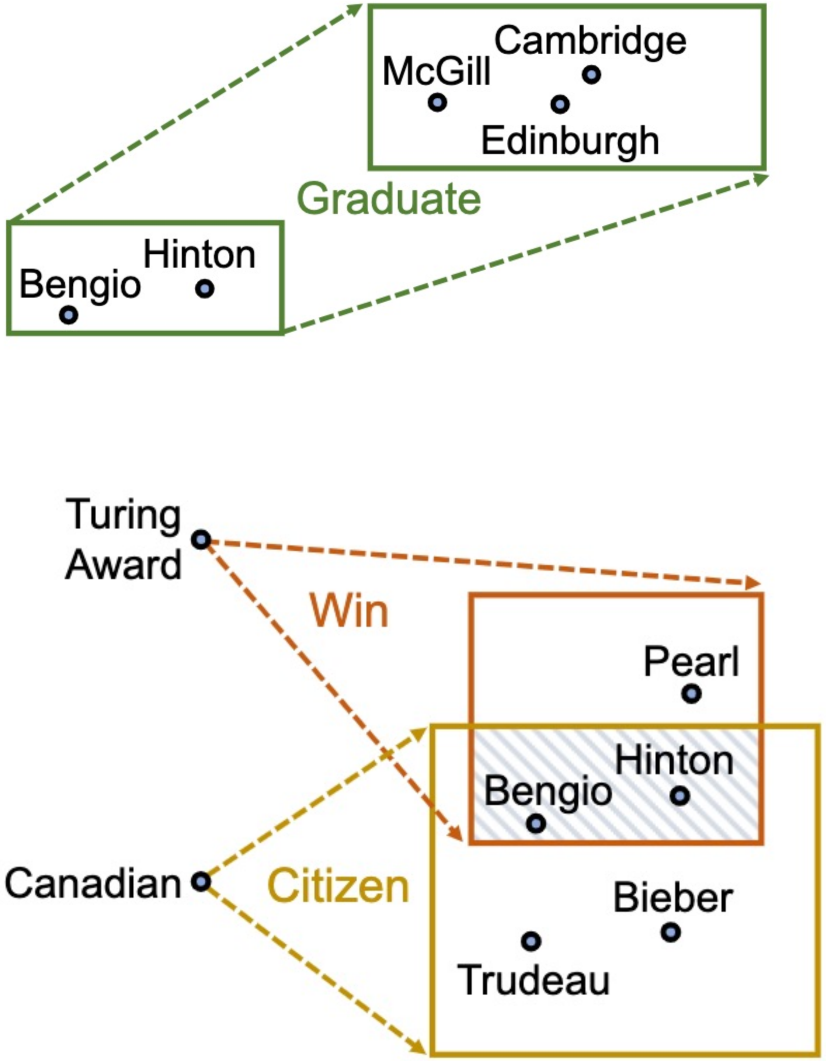
Embedding Space

[Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. Vilnis, et al., ACL 2018]

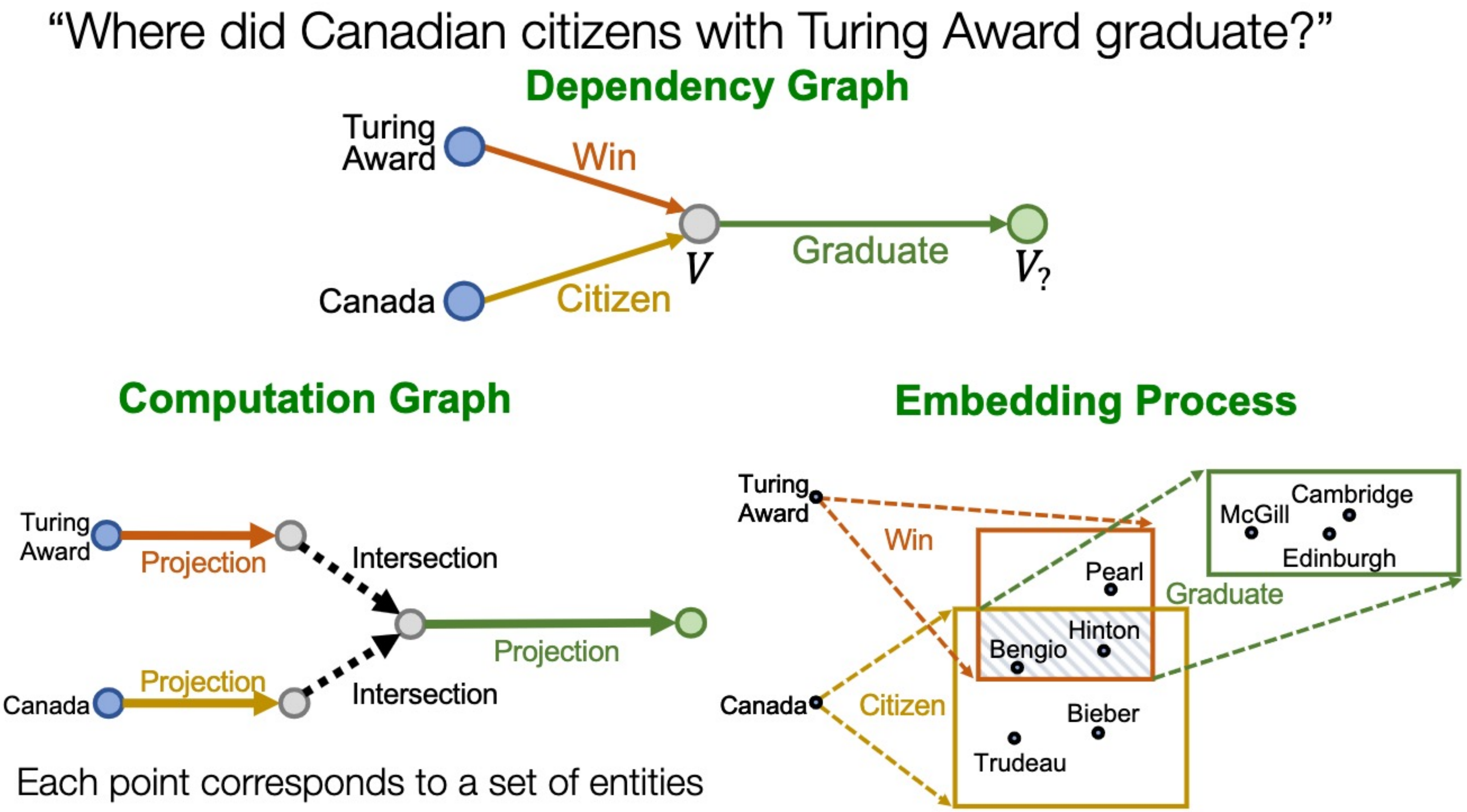
Query2Box



- Geometric Projection Operator
- Geometric Intersection Operator



Query2Box



Each point corresponds to a set of entities

Limitation

- Query2Box can only handle a subset of FOL query
  - e.g., the complement of a box is not a box
- How to handle negation
  - Who are current presidents of European countries which **never** held a (soccer) world cup?
- Why negation is interesting?
  - Real-world reasoning requires negation
  - Full FOL queries
  - Model disjunction

De Morgan's laws: $a \vee b = \neg(\neg a \wedge \neg b)$

BetaE

- **Idea**
  - Embed entities and queries as **Beta distribution**
  - Design probabilistic logical operators for conjunction and negation
- **So that**
  - Negation is represent by **inversing** the shape of distribution
  - Disjunction is represented by

    De Morgan's laws: $a \vee b = \neg(\neg a \wedge \neg b)$

  - Capture uncertainty of query by the entropy of distribution
  - Find answer by measuring the "distance" between query and entities
    - e.g., KL divergence

BetaE

- Basic idea: representing query as beta distribution

$$p(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathbf{B}(\alpha,\beta)}, \text{ where } x \in [0,1] \text{ and } \mathbf{B}(\cdot) \text{ denotes the beta function.}$$

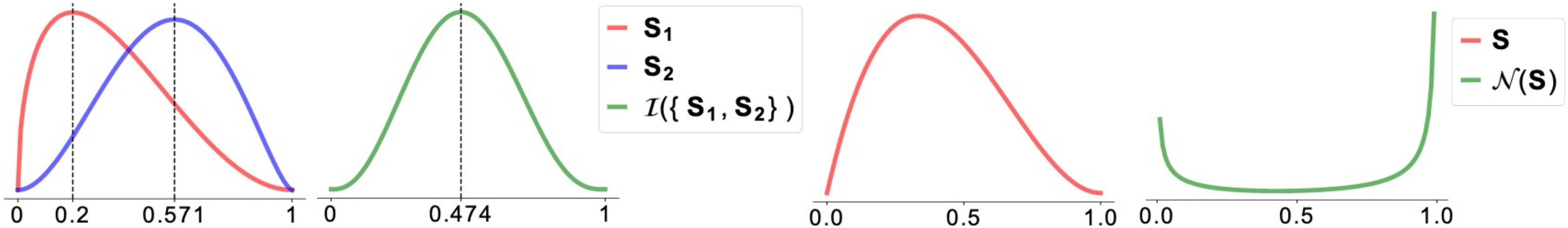


Figure 2: Illustration of our probabilistic intersection operator $\mathcal{I}$ (left) and probabilistic negation operator $\mathcal{N}$ (right). $\mathcal{I}$ transforms the input distribution by taking the weighted product of the PDFs; $\mathcal{N}$ transforms the input distribution by taking the reciprocal of its parameters.

Limitation of BetaE

- BetaE is not fully closed under:
  - **Conjunction,** which defined by weighted product of the PDF of the Beta embeddings
  - **Negation:** which is not the standard complement
- BetaE loses some advantages of geometric embedding

Cone Embedding

- Basic idea: representing query as convex cone

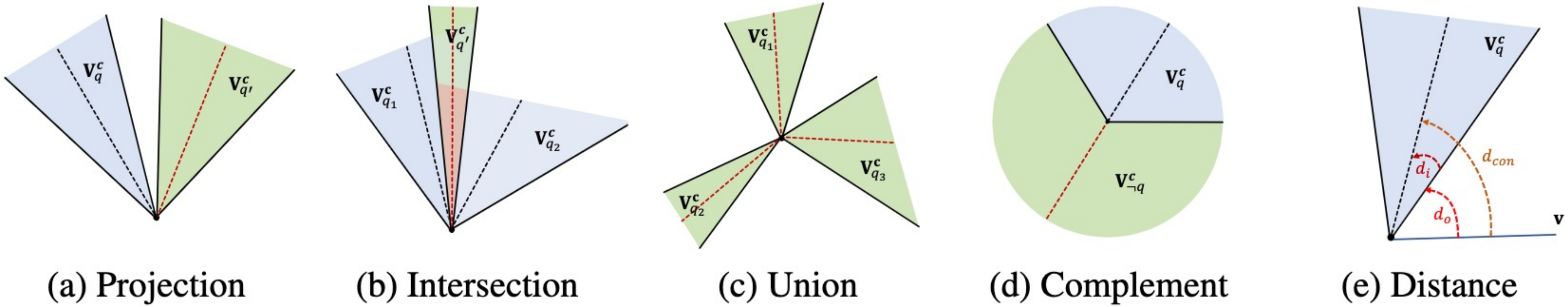


Figure 2: ConE's logical operators and distance function, of which the embedding dimension $d = 1$.

- Only useful in 2d space, and require Cartesian product of multiple 2d space

Results

Table 1: MRR results for answering queries without negation (∃, ∧, ∨) on FB15k, FB237, and NELL. The results of BETAE are taken from Ren & Leskovec [24].

Dataset	Model	1p	2p	3p	2i	3i	pi	ip	2u	up	AVG
FB15k	GQE	53.9	15.5	11.1	40.2	52.4	27.5	19.4	22.3	11.7	28.2
	Q2B	70.5	23.0	15.1	61.2	71.8	41.8	28.7	37.7	19.0	40.1
	BETAE	65.1	25.7	24.7	55.8	66.5	43.9	28.1	40.1	25.2	41.6
	ConE	**73.3**	**33.8**	**29.2**	**64.4**	**73.7**	**50.9**	**35.7**	**55.7**	**31.4**	**49.8**
FB237	GQE	35.2	7.4	5.5	23.6	35.7	16.7	10.9	8.4	5.8	16.6
	Q2B	41.3	9.9	7.2	31.1	45.4	21.9	13.3	11.9	8.1	21.1
	BETAE	39.0	10.9	10.0	28.8	42.5	22.4	12.6	12.4	9.7	20.9
	ConE	**41.8**	**12.8**	**11.0**	**32.6**	**47.3**	**25.5**	**14.0**	**14.5**	**10.8**	**23.4**
NELL	GQE	33.1	12.1	9.9	27.3	35.1	18.5	14.5	8.5	9.0	18.7
	Q2B	42.7	14.5	11.7	34.7	45.8	23.2	17.4	12.0	10.7	23.6
	BETAE	53.0	13.0	11.4	37.6	47.5	24.1	14.3	12.2	8.5	24.6
	ConE	**53.1**	**16.1**	**13.9**	**40.0**	**50.8**	**26.3**	**17.5**	**15.3**	**11.3**	**27.2**

Thank you