Query2Box: Reasoning over Knowledge Graphs in Vector Space

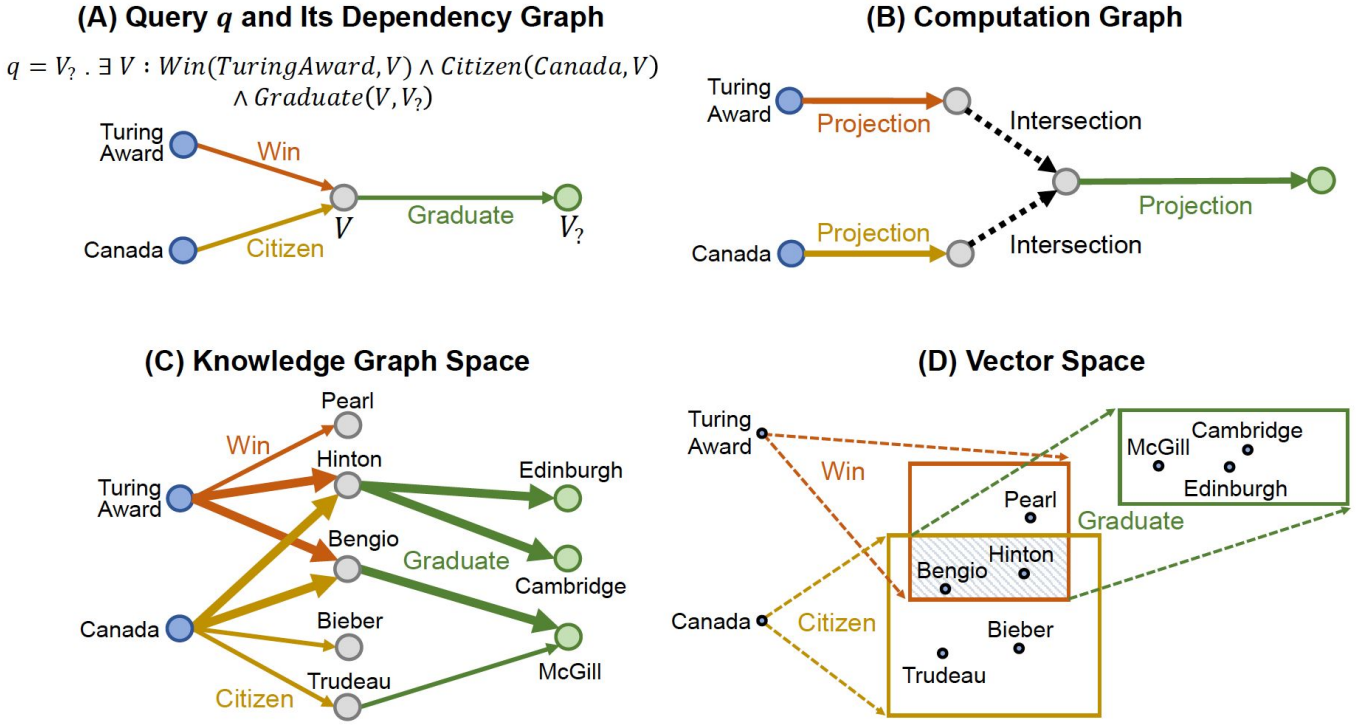
Hongyu Ren, Weihua Hu, Jure Leskovec

Presented by Vardaan Pahuja

Motivation

- An embedding-based framework for reasoning over KGs that is capable of handling arbitrary Existential Positive First-order (EPFO) logical queries
- Key idea: Use a closed region rather than a single point in the vector space.

Query2Box illustration



Query: Where did Canadian citizens with Turing Award graduate

Representing Facts in First-Order Logic

- Lucy* is a professor
- All professors are people.
- John is the dean.
- Deans are professors.
- All professors consider the dean a friend or don't know him
- Everyone is a friend of someone.
- People only criticize people that are not their friends.
- Lucy criticized John .

- is-prof(lucy)
- $\forall$ x (is-prof(x) $\rightarrow$ is-person(x))
- is-dean(John)
- $\forall$ x (is-dean(x) $\rightarrow$ is-prof(x))
- $\forall$ x ($\forall$ y (is-prof(x) $\wedge$ is-dean(y) $\rightarrow$ is-friend-of(y,x) $\vee \neg$ knows(x, y)))
- $\forall$ x ($\exists$ y (is-friend-of (y, x)))
- $\forall$ x ($\forall$ y (is-person(x) $\wedge$ is-person(y) $\wedge$ criticize (x,y) $\rightarrow \neg$ is-friend-of (y,x)))
- criticize(lucy, John)

Slide Credit:
https://www.cs.cornell.edu/courses/cs4700/2011fa/lectures/16_FirstOrderLogic.pdf

Existential Positive First-order logical queries

- Queries that include
  - $Disjunction (\vee)$
  - $Conjunction (\wedge)$
  - $Existential\ quantifier\ (\exists)$

Conjunctive queries

- Queries that include
  - $Conjunction (\wedge)$
  - $Existential\ quantifier\ (\exists)$

$$q = V_? \,.\, \exists V_1, ..., V_m : e_1 \wedge e_2 \wedge ... \wedge e_n,$$
$$\text{where } e_i = \tau(v_j, V_k), V_k \in \{V_?, V_1, ..., V_m\}, v_j \in \mathcal{V}, \tau \in \mathcal{R}$$
$$\text{or } e_i = \tau(V_j, V_k), V_j, V_k \in \{V_?, V_1, ..., V_m\}, j \neq k, \tau \in \mathcal{R}.$$

Notation:

- τ denotes predicates
- $[\![q]\!]$ - answer set or denotation set of query

Problem Statement

- Relational Database: Execute queries on observed edges of graph.
- Goal of this paper: Predict unobserved relationship and not just answer queries that exactly satisfy the query according to observed training edges.
- Train using example query-answer pairs that are known in the training data, i.e., $(q, v^*), v^* \in [\![q]\!]_{\text{train}}$, so that we can generalize to parts of the graph that involve missing edges, i.e., so that we can make predictions for query-answer pairs that rely on edges which are unobserved in the training data $(q, v^*), v^* \in [\![q]\!] \setminus [\![q]\!]_{\text{train}}$

Operators in Computational Graph

- **Projection:** Given a set of entities $S \subseteq \mathcal{V}$, and relation $r \in \mathcal{R}$, this operator obtains $\cup_{v \in S} A_r(v)$, where $A_r(v) \equiv \{v' \in \mathcal{V} : \ r(v, v') = \text{True}\}$.

- **Intersection:** Given a set of entity sets $\{S_1, S_2, \ldots, S_n\}$, this operator obtains $\cap_{i=1}^{n} S_i$.

Logical reasoning in the vector space

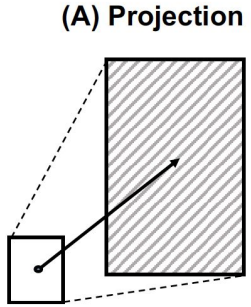
- Model a set of entities in the vector space, we use boxes (i.e., axis-aligned hyper-rectangles).
- The benefit is that unlike a single point, the box has the interior.
- If an entity is in a set, it is natural to model the entity embedding to be a point inside the box.

Formally, we operate on $\mathbb{R}^d$, and define a box in $\mathbb{R}^d$ by $\mathbf{p} = (\mathrm{Cen}(\mathbf{p}), \mathrm{Off}(\mathbf{p})) \in \mathbb{R}^{2d}$ as:

$$\mathrm{Box}_{\mathbf{p}} \equiv \{\mathbf{v} \in \mathbb{R}^d : \mathrm{Cen}(\mathbf{p}) - \mathrm{Off}(\mathbf{p}) \preceq \mathbf{v} \preceq \mathrm{Cen}(\mathbf{p}) + \mathrm{Off}(\mathbf{p})\},$$

- Box parameters:
  - Center of box (denoted by Cen(p))
  - Offset of box (denoted by Off(p))
- Notation:
  - embedding of v is denoted by $\mathbf{v}$
  - the box embedding $\mathbf{p}$ models $\{v \in \mathcal{V} : \mathbf{v} \in \mathrm{Box}_{\mathbf{p}}\}$

Logical reasoning in the vector space



- Start from the initial box embeddings of the source nodes (anchor entities).
  - Each source entity can be modeled as a box centered at the entity vector with zero offset.
- Sequentially update the embeddings according to the logical operators
- **Geometric Projection operator**: Model it using relation embedding for each relation r.
  - $$(\mathrm{Cen}(\mathbf{r}), \mathrm{Off}(\mathbf{r})) \in \mathbb{R}^{2d} \text{ with } \mathrm{Off}(\mathbf{r}) \succeq \mathbf{0}$$
  - Given an input box embedding **p**, the authors model the projection by **p** + **r**, where we sum the centers and sum the offsets
  - This gives us a new box with the translated center and larger offset

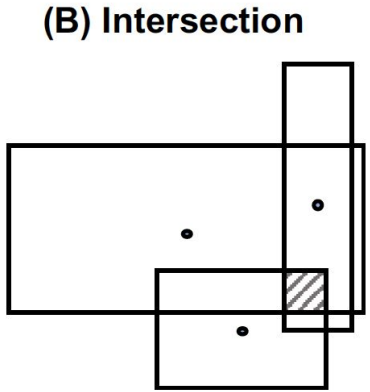
Geometric Intersection Operator

- Center of intersection is obtained by performing attention on the centroids of the individual boxes - the centroid lies inside the set of given boxes.
- Minimum of offsets multiplied by sigmoid gives the resultant offset which is smaller than each of the individual offsets.The
- Intuition behind the geometric intersection is to generate a smaller box that lies inside a set of boxes

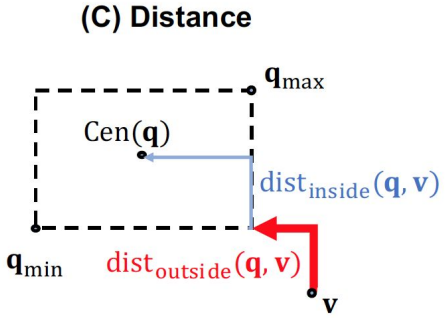
$$\mathrm{Cen}(\mathbf{p}_{\mathrm{inter}}) = \sum_i \mathbf{a}_i \odot \mathrm{Cen}(\mathbf{p_i}), \quad \mathbf{a}_i = \frac{\exp(\mathrm{MLP}(\mathbf{p_i}))}{\sum_j \exp(\mathrm{MLP}(\mathbf{p_j}))},$$

$$\mathrm{Off}(\mathbf{p}_{\mathrm{inter}}) = \mathrm{Min}(\{\mathrm{Off}(\mathbf{p_1}), \ldots, \mathrm{Off}(\mathbf{p_n})\}) \odot \sigma(\mathrm{DeepSets}(\{\mathbf{p_1}, \ldots, \mathbf{p_n}\}))$$

$$\mathrm{DeepSets}(\{\mathbf{x_1}, \ldots, \mathbf{x_N}\}) = \mathrm{MLP}((1/N) \cdot \textstyle\sum_{i=1}^{N} \mathrm{MLP}(\mathbf{x_i}))$$



Entity to Box distance



Entity-to-box distance. Given a query box $\mathbf{q} \in \mathbb{R}^{2d}$ and an entity vector $\mathbf{v} \in \mathbb{R}^{d}$, we define their distance as

$$\text{dist}_{\text{box}}(\mathbf{v}; \mathbf{q}) = \text{dist}_{\text{outside}}(\mathbf{v}; \mathbf{q}) + \alpha \cdot \text{dist}_{\text{inside}}(\mathbf{v}; \mathbf{q}), \tag{3}$$

where $\mathbf{q}_{\max} = \text{Cen}(\mathbf{q}) + \text{Off}(\mathbf{q}) \in \mathbb{R}^{d}$, $\mathbf{q}_{\min} = \text{Cen}(\mathbf{q}) - \text{Off}(\mathbf{q}) \in \mathbb{R}^{d}$ and $0 < \alpha < 1$ is a fixed scalar, and

$$\text{dist}_{\text{outside}}(\mathbf{v}; \mathbf{q}) = \|\text{Max}(\mathbf{v} - \mathbf{q}_{\max}, \mathbf{0}) + \text{Max}(\mathbf{q}_{\min} - \mathbf{v}, \mathbf{0})\|_1,$$
$$\text{dist}_{\text{inside}}(\mathbf{v}; \mathbf{q}) = \|\text{Cen}(\mathbf{q}) - \text{Min}(\mathbf{q}_{\max}, \text{Max}(\mathbf{q}_{\min}, \mathbf{v}))\|_1.$$

- The **outside distance** corresponds to the distance between the entity and closest corner/side of the box.
- The **inside distance** corresponds to the distance between the center of the box and its side/corner (or the entity itself if the entity is inside the box
- The inside distance is down-weighted by using 0 < alpha < 1.
- This means that as long as entity vectors are inside the box, we regard them as “close enough” to the query center

Visualization of geometric operators

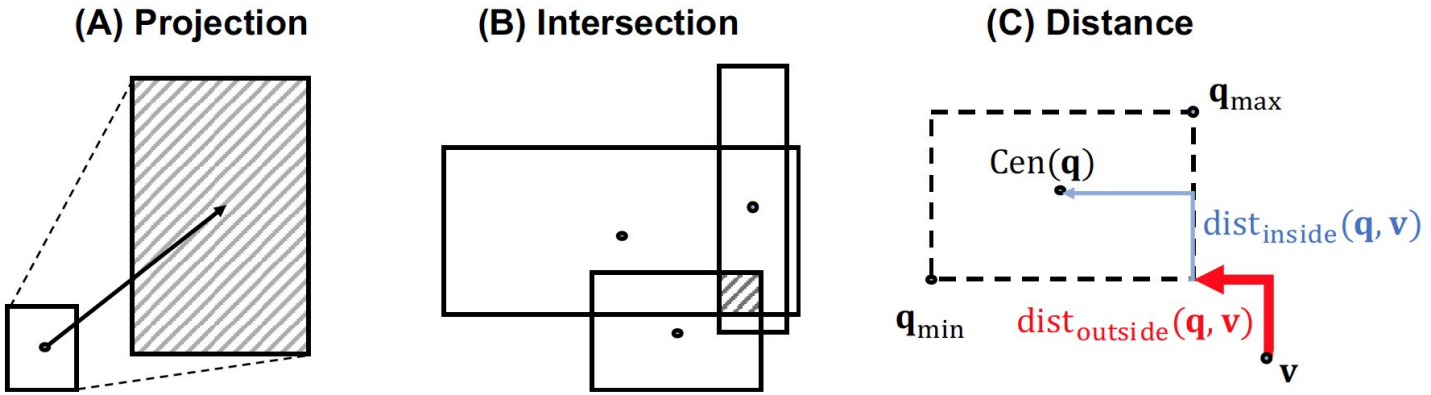


Figure 2: The geometric intuition of the two operations and distance function in QUERY2BOX. **(A)** Projection generates a larger box with a translated center. **(B)** Intersection generates a smaller box lying inside the given set of boxes. **(C)** Distance dist_{box} is the weighted sum of $\text{dist}_{\text{outside}}$ and $\text{dist}_{\text{inside}}$, where the latter is weighted less.

Putting it All Together

- Start from the initial box embeddings of the source nodes (anchor entities).
- Sequentially update the embeddings according to the logical operators.
- This way we will obtain the embedding of the query.
- Answers to the query will be entities that are enclosed in the final query embedding box.

Training Objective

- Given a training set of queries and their answers, we optimize a negative sampling loss to effectively optimize our distance-based model

$$L = -\log \sigma\left(\gamma - \mathrm{dist}_{\mathrm{box}}(\mathbf{v}; \mathbf{q})\right) - \sum_{i=1}^{k} \frac{1}{k} \log \sigma\left(\mathrm{dist}_{\mathrm{box}}(\mathbf{v}'_{\mathbf{i}}; \mathbf{q}) - \gamma\right)$$

where γ represents a fixed scalar margin, $v \in [\![q]\!]$ is a positive entity (*i.e.*, answer to the query q), and $v'_i \notin [\![q]\!]$ is the i-th negative entity (non-answer to the query q) and k is the number of negative entities.

Tractable handling of disjunction

- Existential Positive First-order (EPFO) queries include disjunction in addition to conjunction and existential quantifier (compared to conjunctive queries).
  - **Union:** Given a set of entity sets $\{S_1, S_2, \ldots, S_n\}$, this operator obtains $\cup_{i=1}^{n} S_i$.

- Naive solution: define a geometric operator for union and embed the query
- Challenge: The union of boxes is not a simple box, so such a formulation will make the union operation not closed.

Intractability of Disjunction using Geometric operator

Theorem 1. *Consider any M conjunctive queries $q_1, \ldots, q_M$ whose denotation sets $[\![q_1]\!], \ldots, [\![q_M]\!]$ are disjoint with each other, $\forall\, i \neq j,\ [\![q_i]\!] \cap [\![q_j]\!] = \varnothing$. Let D be the VC dimension of the function class $\{\text{sign}(\beta - \text{dist}(\cdot; \mathbf{q})) : \mathbf{q} \in \Xi\}$, where Ξ represents the query embedding space and $\text{sign}(\cdot)$ is the sign function. Then, we need $D \geq M$ to model any EPFO query, i.e., $\text{dist}(\mathbf{v}; \mathbf{q}) \leq \beta \Leftrightarrow v \in [\![q]\!]$ is satisfied for every EPFO query q.*

- For a real-world KG, there are M ~ |V| conjunctive queries with non-overlapping answers
- Therefore, in order to accurately model **any** EPFO query with the existing framework, the complexity of the distance function measured by the VC dimension needs to be as large as the number of KG entities.
- Thus, the dimensionality of the logical query embeddings needs to be (|V|), which is not low-dimensional and thus not scalable to large KGs

Solution: Transformation to DNF

- Any first-order logic can be transformed into the equivalent DNF
- The transformation is done in the space of computation graph by moving all the edges of type “union” to the last step of the computation graph

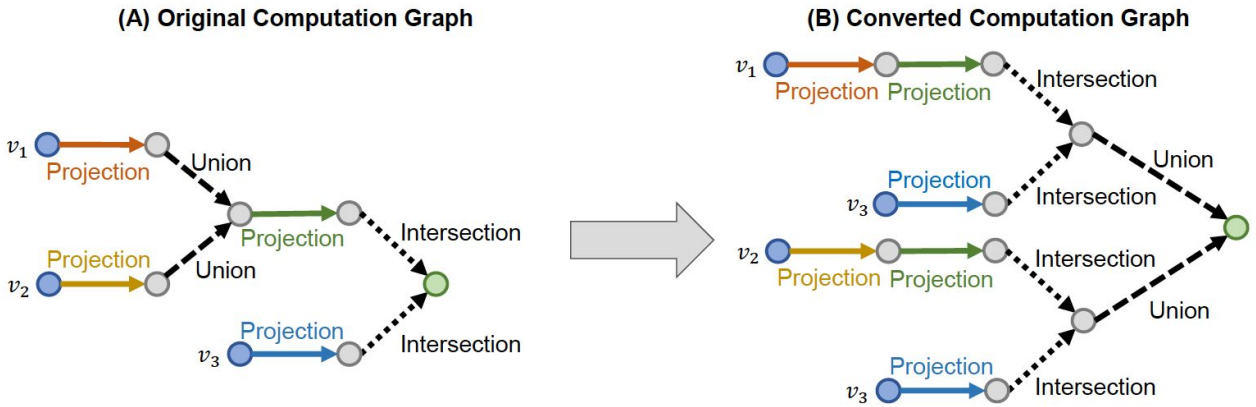


Figure 3: Illustration of converting a computation graph of an EPFO query into an equivalent computation graph of the Disjunctive Normal Form.

Aggregation for DNF queries

$$q \text{ is logically equivalent to } q^{(1)} \vee \cdots \vee q^{(N)}$$

$$\text{dist}_{\text{agg}}(\mathbf{v}; q) = \text{Min}(\{\text{dist}_{\text{box}}(\mathbf{v}; \mathbf{q}^{(1)}), \ldots, \text{dist}_{\text{box}}(\mathbf{v}; \mathbf{q}^{(N)})\})$$

- dist_{agg} takes the minimum distance to the closest box as the distance to an entity.
- This modeling aligns well with the union operation as an entity is inside the union of sets as long as the entity is in one of the sets.
- **Computational complexity**:
  - The computational complexity of answering an EPFO query with this framework is equal to that of answering the N conjunctive queries.
  - It is possible to parallelize these N computations.
- Note that this DNF-query rewriting scheme can be extended any method that works for conjunctive queries e.g., (Hamilton et al., 2018).

Query structures

- Query Structures
  - Training: 5 query structures
  - Evaluation: All 9 query structures.

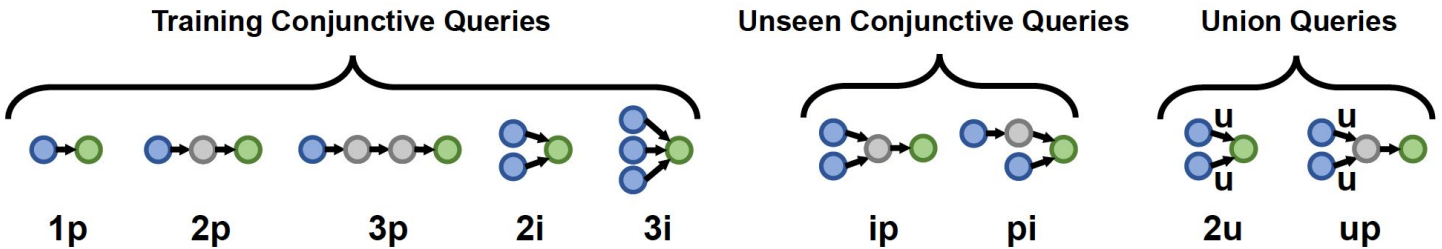


Figure 4: Query structures considered in the experiments, where anchor entities and relations are to be specified to instantiate logical queries. Naming for each query structure is provided under each subfigure, where 'p', 'i', and 'u' stand for 'projection', 'intersection', and 'union', respectively. Models are trained on the first 5 query structures, and evaluated on all 9 query structures. For example, "3p" is a path query of length three, and "2i" is an intersection of cardinality two.

Query Structures Statistics

Dataset	1p	2p	3p	2i	3i	ip	pi	2u	up
FB15k	10.8	255.6	250.0	90.3	64.1	593.8	190.1	27.8	227.0
FB15k-237	13.3	131.4	215.3	69.0	48.9	593.8	257.7	35.6	127.7
NELL995	8.5	56.6	65.3	30.3	15.9	310.0	144.9	14.4	62.5

Table 1: Average number of answer entities of test queries with missing edges grouped by different query structures (for a KG with 10% edges missing).

Query Generation

- Given a KG and a query structure (which is a DAG), we use pre-order traversal to assign an entity and a relation to each node and edge in the DAG of query structure to instantiate a query
- Start from the root of the DAG (which is the target node), we sample an entity **e** uniformly from the KG to be the root
- Then for every node connected to the root in the DAG, we choose a relation **r** uniformly from the in-coming relations of **e** in the KG, and a new entity **e'** from the set of entities that reaches **e** by **r** in the KG.
- Then we assign the relation **r** to the edge and **e'** to the node, and move on the process based on the pre-order traversal

Training splits

- Constructing the split
  - Split edges into training, validation and test sets
  - G_{train} - contains only training edges
  - G_{valid} - contains training + validation edges
  - G_{test} - contains training + validation + test edges
- Notation
  - $[[q]]_{train}$ - Answer set obtained by subgraph matching on G_{train}
  - $[[q]]_{val}$ - Answer set obtained by subgraph matching on G_{valid}
  - $[[q]]_{test}$ - Answer set obtained by subgraph matching on G_{test}
- How train/val/test sets are used
  - Use $[[q]]_{train}$ as positive examples for the query and other random entities as negative examples (Training phase).
  - Evaluate the model on ($[[q]]_{val}$ - $[[q]]_{train}$) to tune hyper-parameters (Validation phase)
  - Report results identifying answer entities on ($[[q]]_{test}$ - $[[q]]_{val}$) (Test phase).
- Focus on answering queries where generalization performance is crucial
  - At least one edge needs to be imputed in order to answer the queries.

Evaluation Metrics

Given a test query q, for each of its non-trivial answers $v \in [\![q]\!]_{\text{test}} \backslash [\![q]\!]_{\text{val}}$, we use dist_{box} in Eq. 3 to rank v among $\mathcal{V} \backslash [\![q]\!]_{\text{test}}$. Denoting the rank of v by $\text{Rank}(v)$, we then calculate evaluation metrics for answering query q, such as Mean Reciprocal Rank (MRR) and Hits at K (H@K):

$$\text{Metrics}(q) = \frac{1}{|[\![q]\!]_{\text{test}} \backslash [\![q]\!]_{\text{val}}|} \sum_{v \in [\![q]\!]_{\text{test}} \backslash [\![q]\!]_{\text{val}}} f_{\text{metrics}}(\text{Rank}(v)), \tag{6}$$

where $f_{\text{metrics}}(x) = \frac{1}{x}$ for MRR, and $f_{\text{metrics}}(x) = \mathbf{1}[x \leq K]$ for H@K.

Datasets

- FB15K
  - 14951 entities
  - 1345 relations
- FB15K-237
  - 14505 entities
  - 237 relations
  - A subset of FB15k where inverse relations are removed.

- NELL995
  - 63361 entities
  - 200 relations
  - A NELL subset that is suitable for multi-hop reasoning from the 995th iteration of the NELL system

Results

Method	Avg	1p	2p	3p	2i	3i	ip	pi	2u	up
FB15k										
Q2B	**0.484**	**0.786**	**0.413**	**0.303**	**0.593**	**0.712**	**0.211**	**0.397**	**0.608**	**0.33**
GQE	0.386	0.636	0.345	0.248	0.515	0.624	0.151	0.310	0.376	0.273
GQE-DOUBLE	0.384	0.630	0.346	0.250	0.515	0.611	0.153	0.320	0.362	0.271
FB15k-237										
Q2B	**0.268**	**0.467**	**0.24**	**0.186**	**0.324**	**0.453**	**0.108**	**0.205**	**0.239**	**0.193**
GQE	0.228	0.402	0.213	0.155	0.292	0.406	0.083	0.17	0.169	0.163
GQE-DOUBLE	0.23	0.405	0.213	0.153	0.298	0.411	0.085	0.182	0.167	0.16
NELL995										
Q2B	**0.306**	**0.555**	**0.266**	**0.233**	**0.343**	**0.48**	**0.132**	**0.212**	**0.369**	**0.163**
GQE	0.247	0.418	0.228	0.205	0.316	0.447	0.081	0.186	0.199	0.139
GQE-DOUBLE	0.248	0.417	0.231	0.203	0.318	0.454	0.081	0.188	0.2	0.139

Table 2: H@3 results of QUERY2BOX vs. GQE on FB15k, FB15k-237 and NELL995.

Discussion

- The model obtains 9.8% (25% relative), 3.8% (15% relative), and 5.9% (24% relative) higher H@3 than the best baselines on FB15k, FB15k-237, and NELL995, respectively.
- Naïvely increasing embedding dimensionality in GQE yields limited performance improvement.
- Q2B performs well on new queries with the same structure as the training queries as well as on new query structures never seen during training, which demonstrates that Q2B generalizes well within and beyond query structures.

Ablation Studies

- Q2B **(our method)**: The box embeddings are used to model queries, and the attention mechanism is used for the intersection operator.
- Q2B-AVG: The attention mechanism for intersection is replaced with averaging.
- Q2B-DEEPSETS: The attention mechanism for intersection is replaced with the deep sets.
- Q2B-AVG-1P: The variant of Q2B-AVG that is trained with only 1p queries (see Fig. 4); thus, logical operators are *not* explicitly trained.
- Q2B-SHAREDOFFSET; The box offset is shared across all queries (every query is represented by a box with the same trainable size).

Ablation Studies

Method	Avg	1p	2p	3p	2i	3i	ip	pi	2u	up
FB15k										
Q2B	**0.484**	0.786	**0.413**	**0.303**	**0.593**	**0.712**	**0.211**	**0.397**	0.608	**0.330**
Q2B-AVG	0.468	0.779	0.407	0.300	0.577	0.673	0.199	0.345	0.607	0.326
Q2B-DEEPSETS	0.467	0.755	0.407	0.294	0.588	0.699	0.197	0.378	0.562	0.324
Q2B-AVG-1P	0.385	**0.812**	0.262	0.173	0.463	0.529	0.126	0.263	**0.653**	0.187
Q2B-SHAREDOFFSET	0.372	0.684	0.335	0.232	0.442	0.559	0.144	0.282	0.417	0.252
FB15k-237										
Q2B	**0.268**	**0.467**	0.24	**0.186**	**0.324**	**0.453**	**0.108**	**0.205**	0.239	**0.193**
Q2B-AVG	0.249	0.462	0.242	0.182	0.278	0.391	0.101	0.158	0.236	0.189
Q2B-DEEPSETS	0.259	0.458	**0.243**	**0.186**	0.303	0.432	0.104	0.187	0.231	0.190
Q2B-AVG-1P	0.219	0.457	0.193	0.132	0.251	0.319	0.083	0.142	**0.241**	0.152
Q2B-SHAREDOFFSET	0.207	0.391	0.199	0.139	0.251	0.354	0.082	0.154	0.15	0.142
NELL995										
Q2B	**0.306**	0.555	**0.266**	**0.233**	**0.343**	**0.480**	**0.132**	**0.212**	0.369	**0.163**
Q2B-AVG	0.283	0.543	0.250	0.228	0.300	0.403	0.116	0.188	0.36	0.161
Q2B-DEEPSETS	0.293	0.539	0.26	0.231	0.317	0.467	0.11	0.202	0.349	0.16
Q2B-AVG-1P	0.274	**0.607**	0.229	0.182	0.277	0.315	0.097	0.18	**0.443**	0.133
Q2B-SHAREDOFFSET	0.237	0.436	0.219	0.201	0.278	0.379	0.096	0.174	0.217	0.137

Table 3: H@3 results of QUERY2BOX vs. several variants on FB15k, FB15k-237 and NELL995.

Ablation Studies

- Importance of attention mechanism
  - Q2B-DEEPSETS is too flexible and neglects the fact that the center should be a weighted average
- Necessity of training on complex queries
  - Training on just 1p queries results in poor accuracy on queries involving logical operations
- Adaptive box size for different queries
  - Different queries have different numbers of answer entities, and the adaptive box size enables us to better model it.

References

- Ren, Hongyu, Weihua Hu, and Jure Leskovec. "Query2box: Reasoning over knowledge graphs in vector space using box embeddings." *arXiv preprint arXiv:2002.05969* (2020).
- Hamilton, Will, et al. "Embedding logical queries on knowledge graphs." *Advances in neural information processing systems*. 2018.
- Xiong, Wenhan, Thien Hoang, and William Yang Wang. "Deeppath: A reinforcement learning method for knowledge graph reasoning." *arXiv preprint arXiv:1707.06690* (2017).

Appendix

- Deep Sets

Theorem 2 *A function $f(X)$ operating on a set X having elements from a countable universe, is a valid set function,* i.e., ***invariant*** *to the permutation of instances in X, iff it can be decomposed in the form $\rho\left(\sum_{x \in X} \phi(x)\right)$, for suitable transformations ϕ and ρ.*