CS6604: Machine Learning on Graphs
Logic Reasoning II

Prayash Joshi, Zhenyu Bi, Meng Lu, Weijie Guan

CS, Virginia Tech

Nov.12, 2024

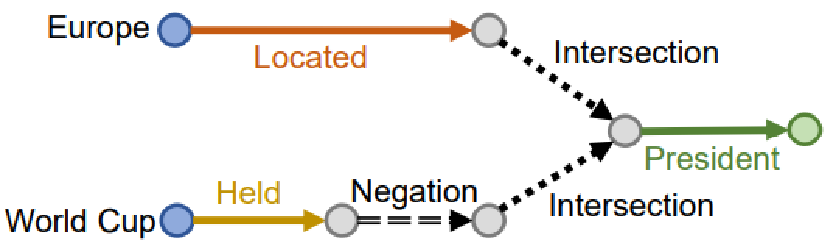
COMPUTER SCIENCE
VIRGINIA TECH

Roadmap

- Background and Motivation
  - Background
  - Research questions
- Algorithms
  - Algorithm #1
  - Algorithm #2
  - Algorithm #3
  - Algorithm #4
- Conclusion

Logical Reasoning

- What is logical reasoning?
  - the process of using structured and coherent thought to analyze arguments, draw conclusions, and make decisions based on evidence and principles of validity
- Why graph logical reasoning?
  - Better Knowledge Representation
  - Inter-displinary Application



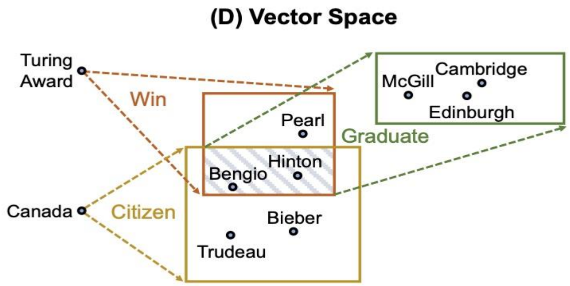
Research Questions

- Q1. How can we represent complex logical queries, and different logical operators?
  - Challenge: Prior work models queries as single points in the vector space, while complex queries could represent many different entities scattered in space.
  - Challenge: Prior work only covers a subset of the query types and operator types.
- Q2. How can we create better embeddings for different logical operators?
  - Challenge: Prior work suffer from model deficiency, which would lead to cascading errors.
- Q3. How can we leverage the ability of LLMs in KG related tasks, such as KGQA?
  - Challenge: LMs cannot interact with KGs well.

Overview

C1: Querie Embedding Representation

Algorithm 1: Query2Box



Algorithm 2: BetaE

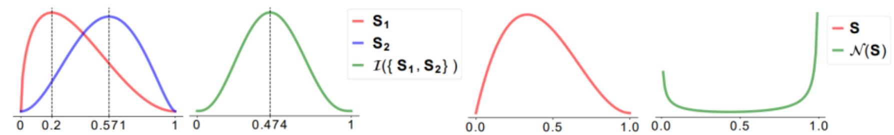
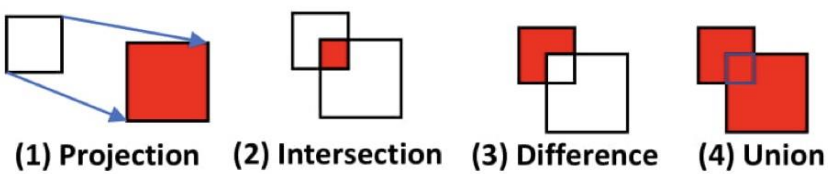


Figure 2: Illustration of our probabilistic intersection operator $\mathcal{I}$ (left) and probabilistic negation operator $\mathcal{N}$ (right). $\mathcal{I}$ transforms the input distribution by taking the weighted product of the PDFs; $\mathcal{N}$ transforms the input distribution by taking the reciprocal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

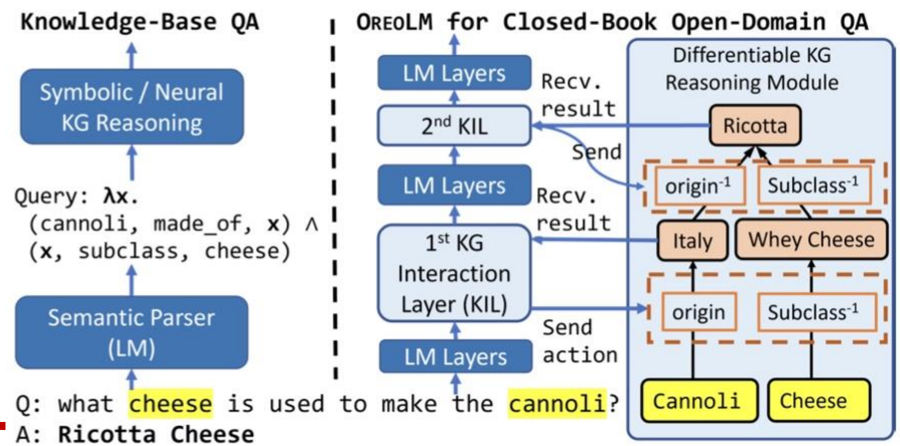
C2: Refined Embedding for Logical Operators

Algorithm 3: NewLook



C3: Language Model with Knowledge Graph

Algorithm 4: OreoLM



Roadmap

- Background and Motivation
  - Background
  - Research questions
- Algorithms
  - Algorithm #1
  - Algorithm #2
  - Algorithm #3
  - Algorithm #4
- Conclusion

Complex
Operators
Disjunctive
Multi-Hop
Machine
edding
Efficiency
Space
Projection
Geometric
Reasoning
Intersection
Relations
Box
EPFO

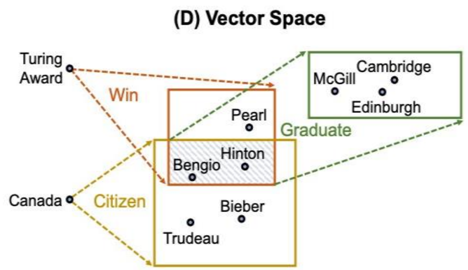
Query2Box

Base
Vector
Normal
Conjunction
Traversal
Graph
AI
Generalization
Existential
Missing
Entity
Form
gical
State-of-the-Art
Quantifiers
Framework
Learning
Accuracy
Disjunction

Hongyu Ren∗ , Weihua Hu∗ , Jure Leskovec
Department of Computer Science, Stanford University
{hyren,weihuahu,jure}@cs.stanford.edu

Key Challenges

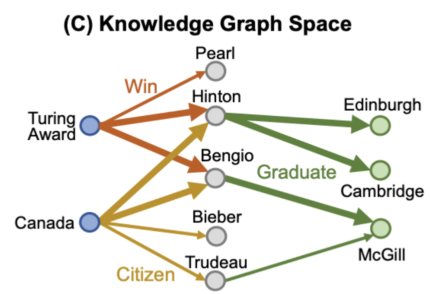
- **Modeling sets in Vector Space**
  - Preceding papers represent queries as single points
    - Fails to capture multiple answer queries
    - Single point not idea for logical operators



- **Handling queries with disjunction and quantifiers**
  - Increases the complexity of reasoning

- **Generalization to new unseen query structure**
  - Models overfit to seen query patterns

- **Scalability to large Incomplete Graphs**
  - Subgraph matching techniques for traditional KGs computationally expensive
  - Sensitive to missing edges



Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings
https://doi.org/10.48550/arXiv.2002.05969

Problem Statement

Given a complex logical query and an incomplete Knowledge Graph,

1. efficiently model and reason over sets of answer entities in a vector space
2. handle arbitrary Existential Positive First-order (EPFO) queries
3. scale to large KGs and generalize to unseen query structures and missing relations

Key Components:

- Box Embeddings
  - *Hyper-rectangles* in vector space
- Geometric Operators
  - *Projections* for relational transitions, *intersections* for modeling logical conjunction
- Disjunctions
  - Query transformation into *Disjunctive Normal Form*

Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings
https://doi.org/10.48550/arXiv.2002.05969

Algorithm

Algorithm 1 Query2Box: Initial Setup of Box Embeddings

Input: Knowledge graph $G = (V, R)$ with entities $v \in V$ and relations $r \in R$
Output: Box embeddings for entities
Initial Setup of Box Embeddings:
for each entity $v \in V$ **do**
    Represent v as a zero-sized box $(v, 0)$ centered at the entity's position
end for

Algorithm 2 Query2Box: Geometric Operations

Input: Relations $r \in R$ with box embeddings and input box $p = (\text{Cen}(p), \text{Off}(p))$
Output: Projected and intersected boxes
Projection Operator:
for each relation $r \in R$ **do**
    Define box embedding $r = (\text{Cen}(r), \text{Off}(r))$
end for
Compute projection:

$$p + r = (\text{Cen}(p) + \text{Cen}(r), \text{Off}(p) + \text{Off}(r))$$

Intersection Operator:
Given multiple boxes $p_1, p_2, \ldots, p_n$
Calculate the center:

$$\text{Cen}(p_{\text{inter}}) = \sum_i a_i \cdot \text{Cen}(p_i), \quad \text{where} \quad a_i = \frac{\exp(\text{MLP}(p_i))}{\sum_j \exp(\text{MLP}(p_j))}$$

Calculate the offset using minimum operation and DeepSets transformation:

$$\text{Off}(p_{\text{inter}}) = \text{Min}(\{\text{Off}(p_1), \ldots, \text{Off}(p_n)\}) \odot \sigma(\text{DeepSets}(\{p_1, \ldots, p_n\}))$$

Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings
https://doi.org/10.48550/arXiv.2002.05969

Algorithm cont'd

Algorithm 3 Query2Box: Distance Function

1: **Input:** Query box q, entity v
2: **Output:** Distance measure between q and v
3: Define distance between query box q and entity v:

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

Algorithm 4 Query2Box: Training Objective and Disjunction Handling

1: **Input:** Query box q, positive entity v, negative samples v'_i
2: **Output:** Trained model and evaluation for disjunctive queries
3: **Training Objective:**
4: Train the model using negative sampling loss:

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

5: Here, v is a positive entity, v'_i are negative samples, γ is a margin, and k is the number of negative samples.
6: **Handling Disjunctions with DNF Transformation:**
7: Convert queries with disjunctions into Disjunctive Normal Form (DNF)
8: For each conjunctive sub-query in DNF, compute separately and aggregate:

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

9: Here, N is the number of conjunctive sub-queries in DNF.

Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings
https://doi.org/10.48550/arXiv.2002.05969

Experiment

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
NELL995	63,361	200	114,213	14,324	14,267	142,804

Table 5: Knowledge graph dataset statistics as well as the split into training, validation, and test sets.

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
NELL995	107,982	107,982	16,927	4,000	17,034	4,000

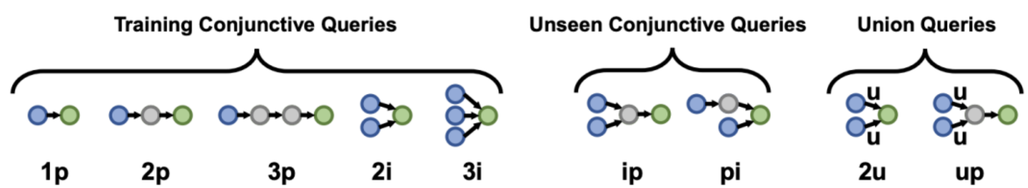
Table 6: Number of training, validation, and test queries generated for different query structu

- Experiments on three standard KG benchmarks (Table 5)
  - FB15k, FB15k-237, and NELL99
- Evaluate on 9 diverse query structures (Figure 4)
  - Train on 5 conjunctive structures
  - Test on 9 unseen conjunctive queries

Goal:

Focus on answering queries requiring multi-hop reasoning over incomplete KGs

- Use Hits@3 and Mean Reciprocal Rank (MRR) as evaluation metrics



Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings
https://doi.org/10.48550/arXiv.2002.05969

Results 1

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.

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

- Q2B achieves higher Hits@3 across all datasets (FB15k, FB15k-237, NELL995) compared to GQE and GQE-DOUBLE.
- Box embeddings effectively capture sets of entities, outperforming point-based embeddings (GQE)
    - Demonstrates strength of box-based modeling in higher embedding dimensions

Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings
https://doi.org/10.48550/arXiv.2002.05969

Results 2

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.

- Q2B with attention-based intersection (Q2B) outperforms variants like Q2B-AVG and Q2B-DEEPSETS
  - Attention in intersection operations shows promising results
- Models with adaptive box sizes (Q2B) yield better results than fixed-size variants (Q2B-SHAREDOFFSET)

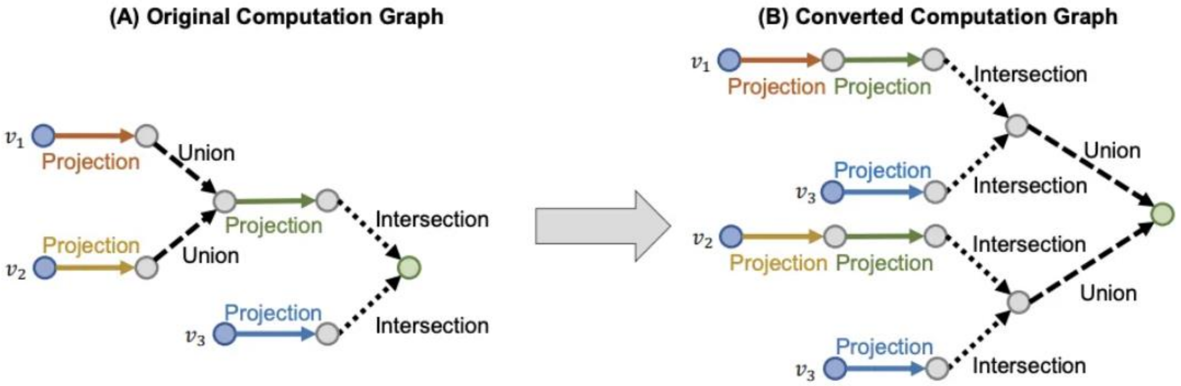
Takeaways

From this paper we got:

- Flexible Box Embedding for Logical Operations
- Improved accuracy on standard benchmarks
    - Robust generalization to unseen queries and multi-hop reasoning tasks

Following papers citing Query2box explored:

- LEGO: Latent Execution-Guided Reasoning for Multi-Hop Question Answering on Knowledge Graphs
- Deep Bidirectional Language-Knowledge Graph Pretraining
- QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering



Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings
https://doi.org/10.48550/arXiv.2002.05969

NEURAL INFORMATION PROCESSING SYSTEMS

2020

Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs

Hongyu Ren, Jure Leskovec
Stanford University

- Ren, H., & Leskovec, J. (2020). Beta embeddings for multi-hop logical reasoning in knowledge graphs. Advances in Neural Information Processing Systems, 33, 19716-19726.

First-order logic queries(FOL)

V is the set of entities. A first-order logic query *q* consists of a non-variable anchor entity set $V_a \subseteq \mathbf{V}$, existentially quantified bound variables $V_1, \ldots, V_k$ and a single target variable $V_?$, which provides the query answer. The disjunctive normal form of a logical query *q* is a disjunction of one or more conjunctions.

$$q[V_?] = V_? \,.\, \exists V_1, \ldots, V_k : c_1 \vee c_2 \vee ... \vee c_n$$

1. Each *c* represents a conjunctive query with one or more literals *e*.

$$c_i = e_{i1} \wedge e_{i2} \wedge \cdots \wedge e_{im}$$

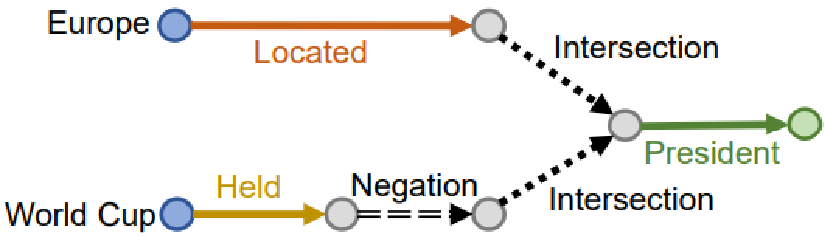
2. Each literal *e* represents an atomic formula or its negation.

$$e_{ij} = r(v_a, V) \; or \; \neg\, r(v_a, V) \; or \; r(V', V) \; or \; \neg\, r(V', V)$$

, where $v_a \in V_a$, $V \in \{V_?, V_1, \ldots, V_k\}$, $V' \in \{V_1, \ldots, V_k\}$, $V \neq V'$, $r \in R$ and each relation type $r \in R$ is a binary function $r : V \times V \rightarrow \{True, False\}$

Computation Graph and Logical Operators

Given a FOL $q = V_? \,.\, \exists V : \textit{Located}(Europe, V) \land \neg \textit{Held}(World\ Cup, V) \land \textit{President}(V, V_?)$, we can derive its corresponding computation graph by representing each atomic formula with relation projection, merging by intersection and transforming negation by complement as follow, which demonstrates the computation process to answer the query



Each node in the computation graph represents a distribution over entities in the KG, and each edge applies a logical operator that transforms this distribution, including operators such as projection, conjunction, disjunction, and negation.

1. **Relation Projection:** Given a set of entities ***S*** $\subseteq$ ***V*** and relation type ***r*** $\in$ ***R***, compute adjacent entities $\cup_{v \in S} A_r(v)$ related to ***S*** via ***r:*** $A_r(v) \equiv \{v' \in \mathcal{V} : r(v, v') = \texttt{True}\}$
2. **Intersection:** Given sets of entities $\{S_1, S_2, \ldots, S_n\}$, compute their intersection $\cap_{i=1}^n S_i$
3. **Complement/Negation:** Given a set of entities ***S*** $\subseteq$ ***V***, compute its complement $\overline{S} \equiv \mathcal{V} \setminus S$
4. **Disjunction/Union:** According to the De Morgan's laws, given sets of $\{S_1, S_2, \ldots, S_n\}$ en $\cup_{i=1}^n S_i$, $\overline{\cap_{i=1}^n \overline{S}}$ is equivalent to

Motivation

Previous works only support existential positive first-order (EPFO) queries, a subset of FOL queries with existential quantification (∃), conjunction (∧) and disjunction (∨), but not negation (¬) because these methods embed queries as closed regions.

However, negation is a fundamental operation and required for the complete set of FOL operators.

Object of this work:

Defining a probabilistic embedding framework for answering arbitrary First-order logic queries queries over KGs which can handle a complete set of first-order logical operations: conjunction (∧), disjunction (∨), and negation (¬)

Beta Embedding

In order to define negation/complement, the authors propose to embed both the entities and queries into the same space using probabilistic embeddings with bounded support. And Beta distribution Beta(α, β) which has two shape parameters is able to modeling a range of probability densities within a bounded interval [0, 1]. The density function of Beta distribution is $p(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathbf{B}(\alpha,\beta)}$, where x ∈ [0, 1] and B(·) denotes the beta function.

A Beta Embedding consists of multiple independent Beta distributions which can be represent with the parameter of each beta distribution as $S = [(\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n)]$, where n is a hyperparameter for the dimension.

1. **For entities**, each entity are viewed as a set with a single element and assigned an initial Beta embedding with learnable parameters.

2. **For queries**, each query is embedded as a Beta embedding which is calculated by a set of probabilistic logical operators following the computation graph

Properties of Beta Embedding

1. **Naturally model uncertainty**

The uncertainty of a Beta distribution can be measured by its differential entropy as

$$u = H = \ln B(\alpha, \beta) - (\alpha - 1)[\psi(\alpha) - \psi(\alpha + \beta)] - (\beta - 1)[\psi(\beta) - \psi(\alpha + \beta)]$$

where $\psi(\cdot)$ represents the digamma function

which can be applied directly to estimate the uncertainty for Beta Embedding

2. **Logical/Set operators (conjunction/intersection and especially negation/complement) are closed**

   1. Given Beta embedding S, S is a fixed point of $\mathcal{N} \circ \mathcal{N}$: $\mathcal{N}(\mathcal{N}(\mathbf{S})) = \mathbf{S}$

   2. Given Beta embedding S, we have $\mathcal{I}(\{\mathbf{S}, \mathbf{S}, \ldots, \mathbf{S}\}) = \mathbf{S}$

Logical Operators

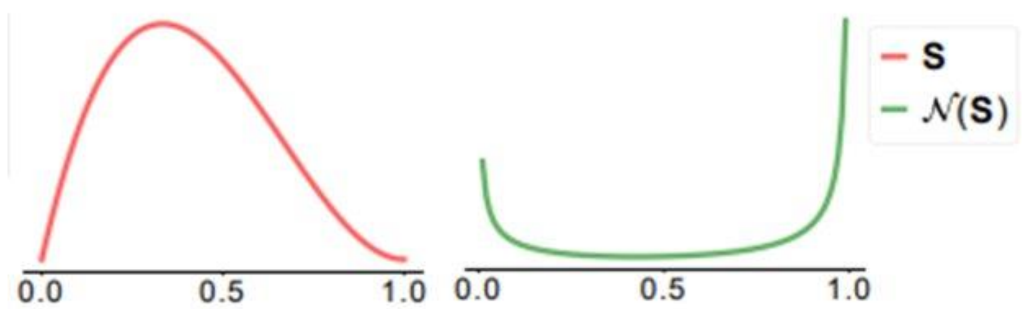
By embedding both the entities and queries into the same space by assigning initial Beta embeddings with learnable parameters $\mathbf{S} = [(\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n)]$ ere n is a hyperparameter, the Logical Operators can be formulated as follow

Probabilistic Projection Operator P

Mapping S to S' with relationship r by $\mathbf{S}' = \texttt{MLP}_r(\mathbf{S})$

Probabilistic Negation Operator N

Negation Operator can be defined as $\mathcal{N}([(\alpha, \beta)]) = [(\frac{1}{\alpha}, \frac{1}{\beta})]$



Logical Operators

Probabilistic Intersection Operator I

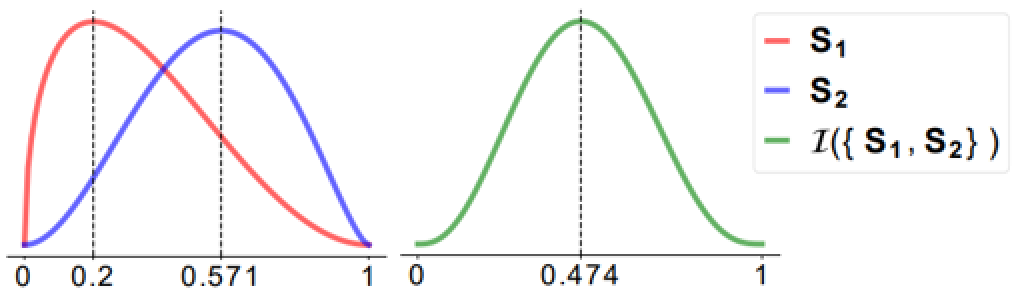
Given n input embeddings $\{\mathbf{S_1}, \ldots, \mathbf{S_n}\}$, Intersection Operator I can be defined by taking the weighted product of the PDFs of the input Beta embeddings

$$p_{\mathbf{S}_{\mathrm{Inter}}} = \frac{1}{Z} \prod p_{\mathbf{S_1}}^{w_1} \cdots p_{\mathbf{S_n}}^{w_n}$$

Where p_S is the Beta distribution PDF for embedding S, Z is a normalization constant and $w_1, \cdots, w_n$ are the weights with their sum equal to 1 which are calculated as follow

$$w_i = \frac{\exp(\mathrm{MLP}_{\mathrm{Att}}(\mathbf{S_i}))}{\sum_j \exp(\mathrm{MLP}_{\mathrm{Att}}(\mathbf{S_j}))}$$

With the property of Beta distribution, we can derive $\mathbf{S}_{\mathrm{Inter}}$ as $[(\sum w_i\alpha_i, \sum w_i\beta_i)]$ when S = $[(\alpha_1, \beta_1)]$



Training Objective

Training Target

Minimize the distance between the Beta embedding of a query and its answers while maximizing the distance between the Beta embedding of the query and other random entities via negative sampling

$$L = -\log \sigma \left(\gamma - \mathtt{Dist}(v;q)\right) - \sum_{j=1}^{k} \frac{1}{k} \log \sigma \left(\mathtt{Dist}(v'_j;q) - \gamma\right)$$

And the distance between entity v and query q is defined as

$$\mathtt{Dist}(v;q) = \sum_{i=1}^{n} \mathrm{KL}(p_{\mathbf{v},\mathbf{i}}; p_{\mathbf{q},\mathbf{i}}),$$

Experiments: QA

Dataset	Model	1p	2p	3p	2i	3i	pi	ip	2u		up		avg
									DNF	DM	DNF	DM	
FB15k	BetaE	65.1	**25.7**	**24.7**	**55.8**	**66.5**	**43.9**	**28.1**	**40.1**	25.0	**25.2**	25.4	**41.6**
	Q2B	**68.0**	21.0	14.2	55.1	**66.5**	39.4	26.1	35.1	-	16.7	-	38.0
	GQE	54.6	15.3	10.8	39.7	51.4	27.6	19.1	22.1	-	11.6	-	28.0
FB15k-237	BetaE	39.0	**10.9**	**10.0**	28.8	**42.5**	**22.4**	**12.6**	**12.4**	11.1	**9.7**	9.9	**20.9**
	Q2B	**40.6**	9.4	6.8	**29.5**	42.3	21.2	**12.6**	11.3	-	7.6	-	20.1
	GQE	35.0	7.2	5.3	23.3	34.6	16.5	10.7	8.2	-	5.7	-	16.3
NELL995	BetaE	**53.0**	13.0	**11.4**	**37.6**	**47.5**	**24.1**	14.3	**12.2**	11.0	8.5	8.6	**24.6**
	Q2B	42.2	**14.0**	11.2	33.3	44.5	22.4	**16.8**	11.3	-	**10.3**	-	22.9
	GQE	32.8	11.9	9.6	27.5	35.2	18.4	14.4	8.5	-	8.8	-	18.6

Table 1: MRR results (%) of BetaE, Q2B and GQE on answering EPFO ($\exists$, $\wedge$, $\vee$) queries.

PS: p projection, i intersection, n negation, u union

Experiments: Uncertainty vs. Answer Size

Dataset	Model	1p	2p	3p	2i	3i	pi	ip	2in	3in	inp	pin	pni
FB15k	Q2B	0.301	0.219	0.262	0.331	0.270	0.297	0.139	-	-	-	-	-
	BETAE	**0.373**	**0.478**	**0.472**	**0.572**	**0.397**	**0.519**	**0.421**	0.622	0.548	0.459	0.465	0.608
FB15k-237	Q2B	0.184	0.226	0.269	0.347	0.436	0.361	0.199	-	-	-	-	-
	BETAE	**0.396**	**0.503**	**0.569**	**0.598**	**0.516**	**0.540**	**0.439**	0.685	0.579	0.511	0.468	0.671
NELL995	Q2B	0.154	0.288	0.305	0.380	0.410	0.361	0.345	-	-	-	-	-
	BETAE	**0.423**	**0.552**	**0.564**	**0.594**	**0.610**	**0.598**	**0.535**	0.711	0.595	0.354	0.447	0.639

Table 3: Spearman's rank correlation between learned embedding (differential entropy for BETAE, box size for Q2B) and the number of answers of queries. BETAE shows up to 77% relative improvement.

1p	2p	3p	2i	3i	pi	ip	2in	3in	inp	pin	pni
0.825	0.766	0.793	0.909	0.933	0.868	0.798	0.865	0.93	0.801	0.809	0.848

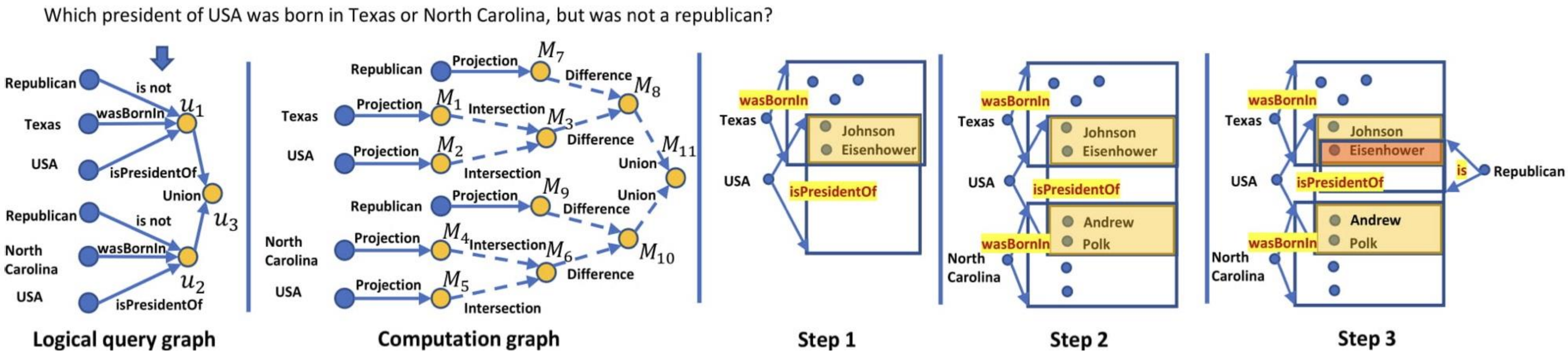
Table 4: ROC-AUC score of BETAE for all the 12 query structures on classification of queries with/without answers on the NELL dataset.

Neural-Answering Logical Queries on Knowledge Graphs

Lihui Liu, Boxin Du, Heng Ji, ChengXiang Zhai, Hanghang Tong
Department of Computer Science, University of Illinois at Urbana Champaign

- Liu, L., Du, B., Ji, H., Zhai, C., & Tong, H. (2021). Neural-Answering Logical Queries on Knowledge Graphs. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Background



- Query graph
  - We can transform a question into a logical query graph
  - e.g.: Who is the spouse of Obama -> ? ---(is married to)---> Obama
- Logical Operations
  - Projection, Intersection, Difference, Union
  - Operating on box embeddings

Challenges and Previous Work

- Previously: Sub-graph matching
  - o Real-word graphs are often incomplete
  - o Empty answer, wrong answer, high computing time
- Recently: Embedding-based methods
  - o Cascade of errors, especially in long and complex queries
  - o Limited support of operators

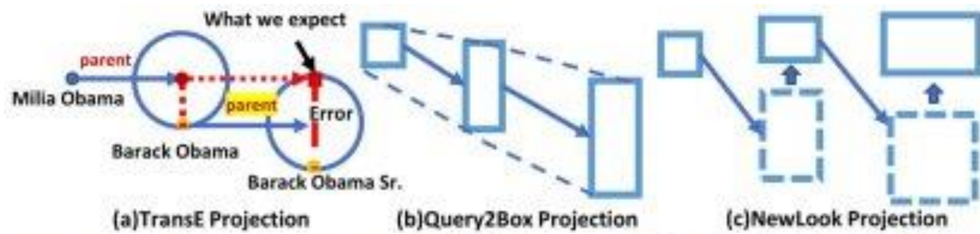
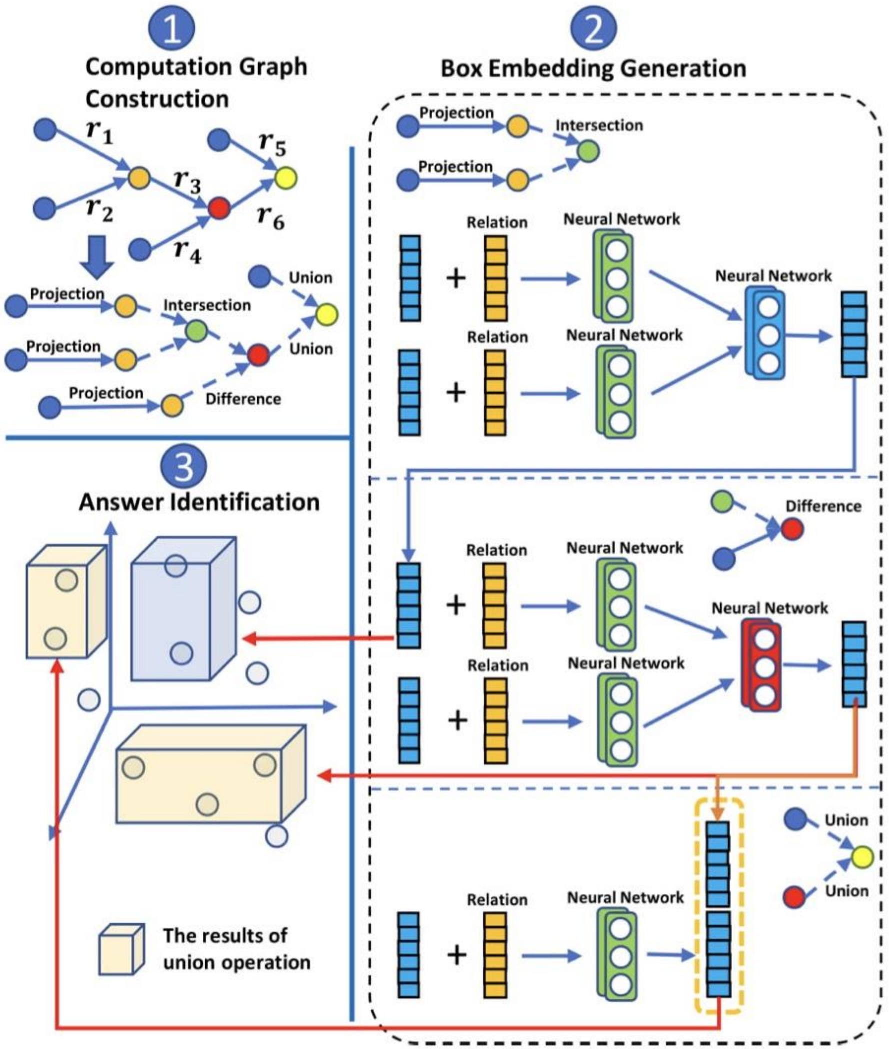
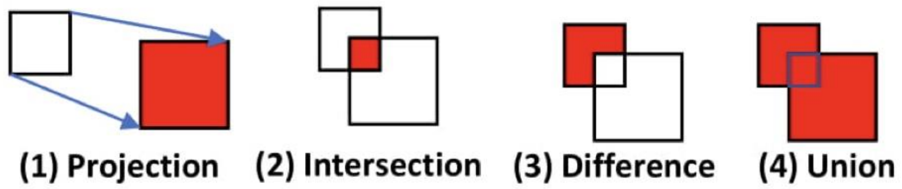
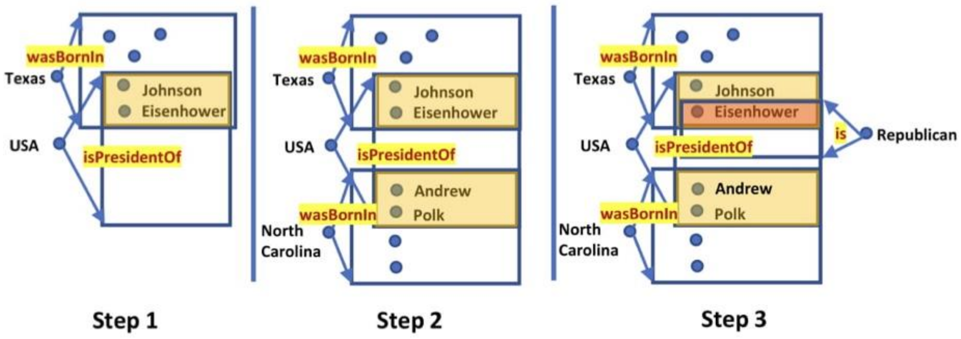


Figure 5: Comparison of different methods for modeling projection operation. Both TransE (a) and Query2Box (b) suffer from severe cascading errors. The proposed NewLook (c) models the projection as a linear transformation (the dashed boxes), followed by a nonlinear process (the solid boxes above the corresponding dashed boxes). NewLook is able to mitigate the cascading errors by adaptively adjusting the center and offset of the projection box.

Operators: What and How







Projection

- Instead of modeling projection as linear transformation, this paper uses neural network to learn the projection operation
- Adaptively adjust the center and offset of the box, which which mitigates the cascading error problem.

$$
\begin{aligned}
\hat{\mathbf{b}}_t^c &= \mathbf{b}_h^c + \mathbf{p}_r^c & \hat{\mathbf{b}}_t^o &= \mathbf{p}_r^o \\
\mathbf{z}_1 &= \mathrm{MLP}(\hat{\mathbf{b}}_t^c) & \mathbf{z}_2 &= \mathrm{MLP}(\hat{\mathbf{b}}_t^o) \\
\mathbf{b}_t^c &= \mathrm{MLP}(\mathbf{z}_1 || \mathbf{z}_2 || \mathbf{x}_{u_t}) & \mathbf{b}_t^o &= \mathrm{MLP}(\mathbf{z}_1 || \mathbf{z}_2 || \mathbf{x}_{u_t})
\end{aligned}
\tag{2}
$$

Intersection and Difference

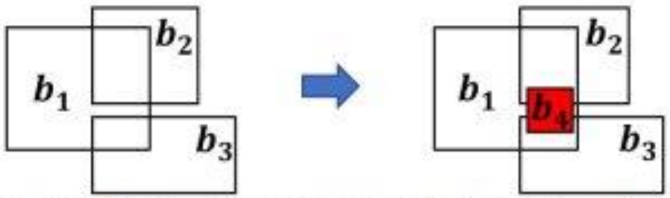


Figure 6: An illustration of intersection operation. Given three boxes b_1, b_2 and b_3 on the left, whose intersection is empty in the training set but not in the ground truth. A neural network approach based on attention mechanism and deepset can result in a reasonable box b_4 on the right.

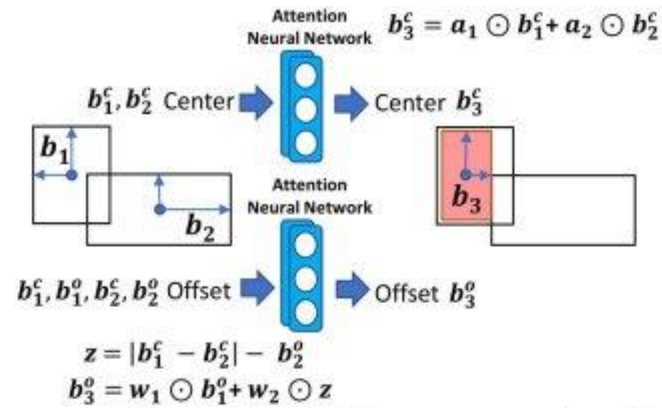


Figure 7: An illustration of difference operation. The proposed NEWLOOK uses two attention networks to learn the center and offset of the box embedding b_3, with the help of z to indicate if and how the two input boxes are overlapped with each other.

$$
\begin{aligned}
\mathbf{z}_i &= \frac{1}{\left\|\mathrm{Relu}(\mathbf{x}_{u_i} - \mathbf{x}_{u_t})\right\|_1} & \hat{\mathbf{b}}^o &= \mathrm{Min}(\{\mathbf{b}_1^o, \ldots, \mathbf{b}_k^o\}) \\
\mathbf{a}_i &= \frac{\exp(\mathbf{z}_i \mathrm{MLP}(\mathbf{b}_i^c || \mathbf{b}_i^o))}{\sum_{j=1}^{k} \exp(\mathbf{z}_k \mathrm{MLP}(\mathbf{b}_k^c || \mathbf{b}_k^o))} & \mathbf{b}_t^c &= \sum_{i=1}^{k} \mathbf{a}_i \odot \mathbf{b}_i^c \qquad (3) \\
\mathbf{w} &= \mathrm{MLP}(\mathrm{Mean}(\sum_{i=1}^{k} \mathrm{MLP}(\mathbf{b}_i^c || \mathbf{b}_i^o))) & \mathbf{b}_t^o &= \hat{\mathbf{b}}^o \odot \sigma(\mathbf{w})
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{t} &= \exp(\mathrm{MLP}(\mathbf{b}_1^o)) & \mathbf{a}_i &= \frac{\exp(\mathrm{MLP}(\mathbf{b}_i^c)))}{\sum_{i=1}^{k} \exp(\mathrm{MLP}(\mathbf{b}_i^c))} \\
\mathbf{z}_i &= |\mathbf{b}_1^c - \mathbf{b}_i^c| + \mathbf{b}_1^o - \mathbf{b}_i^o \;\; (i = 2, .., k) & & \\
\mathbf{w}_i &= \frac{\exp(\mathrm{MLP}(\mathbf{z}_i))}{\mathbf{t} + \sum_{i=2}^{k} \exp(\mathrm{MLP}(\mathbf{z}_i))} & \mathbf{b}_t^c &= \sum_{i=1}^{k} \mathbf{a}_k \odot \mathbf{b}_k^c \qquad (4) \\
\mathbf{w}_1 &= \frac{\mathbf{t}}{\mathbf{t} + \sum_{i=2}^{k} \exp(\mathrm{MLP}(\mathbf{z}_i))} & \mathbf{b}_t^o &= \sum_{i=1}^{k} \mathbf{w}_i \odot \mathbf{b}_i^o
\end{aligned}
$$

Training

- Generate a set of queries together with their answers, and then learn entity embedding, relation em- bedding and geometric logical operations at the same time
- Use negative sampling to optimize the model

$$L = \sum_{u_j \in U_?} [-\log \sigma(\gamma - d(v, \mathbf{b}_j) - \lambda * ||\mathbf{Relu}(\mathbf{x}_v - \mathbf{x}_{u_j})||_1) \\ - \frac{1}{n} \sum_{m=1}^{n} \log \sigma(\lambda * ||\mathbf{Relu}(\mathbf{x}_{m'} - \mathbf{x}_{u_j})||_1 + d(v_{m'}, \mathbf{b}_j) - \gamma)] \quad (5)$$

Dataset and Evaluation Metrics

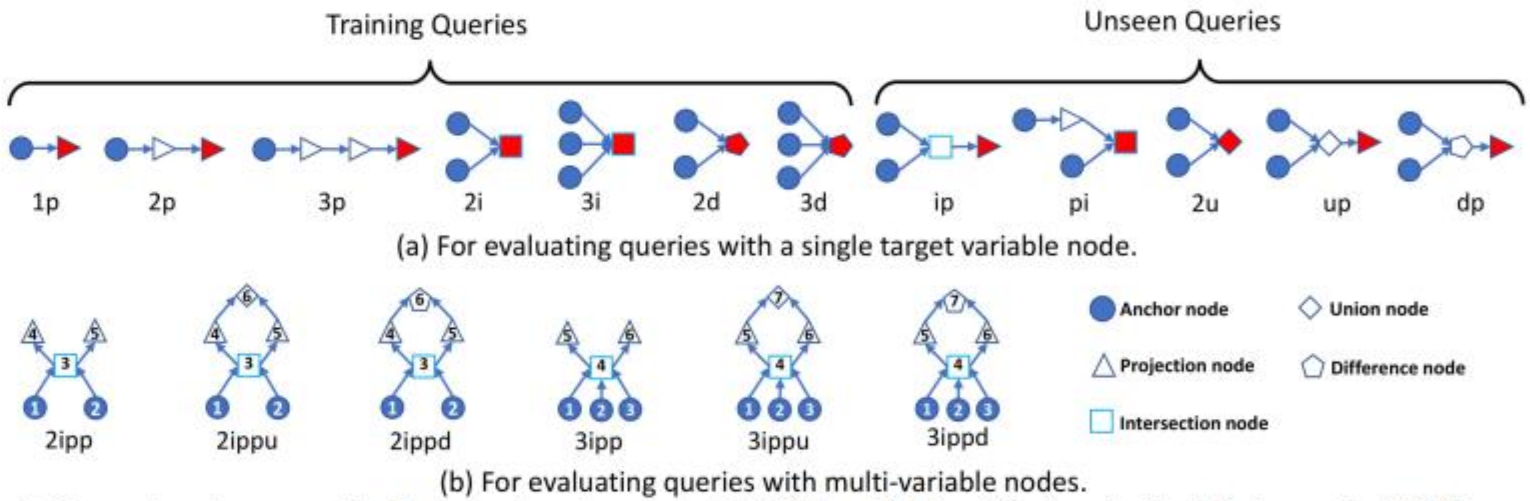


Figure 8: Query structures used in the experiments, where 'p', 'i', 'd' and 'u' stand for 'projection', 'intersection', 'difference' and 'union', respectively. Numbers before 'i', 'd' and 'u' denote their input size (i.e., the number of anchor nodes). Number before 'p' denotes the length of the path (i.e., the number of projection operations).

- Dataset: FB15K, FB15K-237, NELL
- Metric: Hit@k

$$\mathrm{H@K}(q) = \frac{1}{|A_q|} \sum_{v \in A_q} \mathbb{1}(\mathrm{Rank}(v) \leq K)$$

Experiment Results

Table 2: Answering queries with a single target variable node. NLK refers to NewLook, Q2B refers to Query2Box.

Query	1p			2p			3p			2i			3i			ip		
Method	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk
FB15k																		
Hits@1	0.31	0.48	**0.81**	0.15	0.23	**0.51**	0.11	0.15	**0.39**	0.20	0.32	**0.51**	0.27	0.41	**0.69**	0.07	0.10	**0.21**
Hits@3	0.69	0.78	**0.88**	0.32	0.38	**0.64**	0.22	0.26	**0.51**	0.44	0.58	**0.72**	0.55	0.69	**0.78**	0.13	0.18	**0.31**
Hits@10	0.85	0.90	**0.93**	0.48	0.54	**0.75**	0.35	0.40	**0.64**	0.63	0.75	**0.80**	0.74	0.84	**0.89**	0.24	0.30	**0.43**
FB15k-237																		
Hits@1	0.20	0.27	**0.68**	0.10	0.13	**0.30**	0.07	0.09	**0.19**	0.10	0.14	**0.47**	0.16	0.22	**0.57**	0.04	0.05	**0.08**
Hits@3	0.39	0.45	**0.85**	0.19	0.22	**0.43**	0.13	0.16	**0.31**	0.25	0.31	**0.71**	0.36	0.43	**0.71**	0.07	0.10	**0.16**
Hits@10	0.57	0.63	**0.94**	0.32	0.36	**0.59**	0.24	0.28	**0.45**	0.43	0.50	**0.81**	0.54	0.61	**0.81**	0.15	0.19	**0.25**
NELL																		
Hits@1	0.13	0.20	**0.81**	0.08	0.11	**0.45**	0.08	0.10	**0.33**	0.09	0.14	**0.60**	0.14	0.24	**0.66**	0.04	0.05	**0.13**
Hits@3	0.44	0.53	**0.92**	0.20	0.23	**0.34**	0.19	0.21	**0.48**	0.27	0.33	**0.74**	0.36	0.45	**0.80**	0.08	0.11	**0.21**
Hits@10	0.62	0.70	**0.96**	0.35	0.39	**0.48**	0.30	0.34	**0.63**	0.47	0.55	**0.84**	0.58	0.66	**0.89**	0.17	0.20	**0.32**

Query	pi			2u			up			2d			3d			dp			Average		
Method	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk
FB15k																					
Hits@1	0.12	0.19	**0.48**	0.14	0.25	**0.81**	0.11	0.17	**0.25**	0.31	0.49	**0.88**	0.12	0.15	**0.38**	0.17	0.23	**0.34**	0.17	0.27	**0.52**
Hits@3	0.27	0.37	**0.61**	0.43	0.59	**0.94**	0.24	0.29	**0.37**	0.66	0.77	**0.95**	0.36	0.32	**0.54**	0.36	0.33	**0.46**	0.39	0.47	**0.64**
Hits@10	0.44	0.54	**0.74**	0.68	0.81	**0.98**	0.40	0.46	**0.51**	0.83	0.90	**0.97**	0.44	0.46	**0.69**	0.52	0.46	**0.59**	0.56	0.62	**0.74**
FB15k-237																					
Hits@1	0.06	0.09	**0.28**	0.05	0.07	**0.49**	0.06	0.09	**0.15**	0.29	0.40	**0.76**	0.16	0.26	**0.29**	0.15	0.16	**0.27**	0.12	0.16	**0.37**
Hits@3	0.15	0.19	**0.41**	0.15	0.22	**0.70**	0.14	0.17	**0.24**	0.54	0.64	**0.86**	0.38	0.44	**0.46**	0.28	0.26	**0.39**	0.25	0.29	**0.52**
Hits@10	0.27	0.32	**0.54**	0.33	0.43	**0.86**	0.27	0.31	**0.37**	0.72	0.79	**0.94**	0.57	0.62	**0.64**	0.43	0.39	**0.53**	0.39	0.45	**0.64**
NELL																					
Hits@1	0.08	0.10	**0.33**	0.03	0.07	**0.68**	0.04	0.06	**0.23**	0.20	0.29	**0.88**	0.16	0.28	**0.36**	0.19	0.19	**0.46**	0.11	0.14	**0.49**
Hits@3	0.17	0.19	**0.46**	0.22	0.33	**0.85**	0.12	0.13	**0.35**	0.55	0.69	**0.95**	0.41	0.43	**0.54**	0.38	0.33	**0.60**	0.28	0.33	**0.60**
Hits@10	0.28	0.32	**0.59**	0.44	0.56	**0.93**	0.27	0.29	**0.47**	0.72	0.82	**0.98**	0.63	0.64	**0.71**	0.54	0.49	**0.71**	0.45	0.49	**0.71**

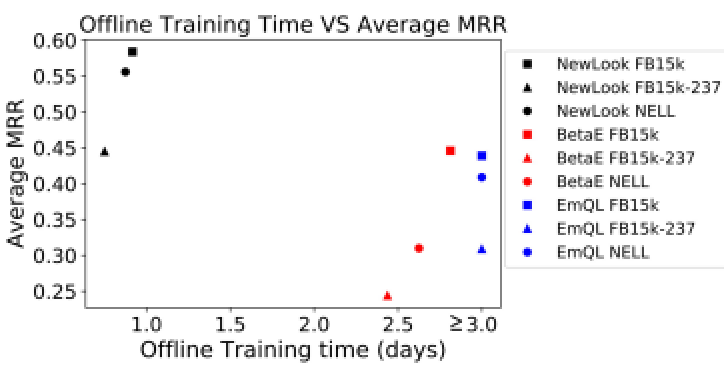


Figure 9: Average MRR results on the Query2Box datasets.

Table 3: Answering queries with multi-variable nodes.

Method	GQE	Q2B	GRay	FilM	GFinder	NewLook
2ipp	0.550	0.481	0.554	0.566	0.638	**0.720**
2ippu	0.592	0.426	0.505	0.583	0.631	**0.761**
2ippd	0.462	0.435	0.452	0.637	**0.652**	0.641
3ipp	0.447	0.442	0.513	0.394	0.437	**0.688**
3ippu	0.507	0.417	0.456	0.408	0.465	**0.733**
3ippd	0.447	0.395	0.421	0.423	0.482	**0.634**
Average	0.500	0.432	0.483	0.501	0.550	**0.696**

Background

- **ODQA Overview**: Requires models to answer questions beyond the immediate context.
- **Challenges with Traditional Language Models**: Closed-Book Models lack external knowledge coverage.
- **Advantages of Knowledge Graphs (KGs)**:

    1. Compact Knowledge Representation: Triples offer efficient storage and retrieval.

    2. Logical Reasoning: KGs allow for path-based reasoning to fill knowledge gaps.

- **Current Limitations in KG and LM Integration**:
  - Most methods use LMs as parsers, limiting deep interaction and multi-hop reasoning.

- Paper: **Empowering Language Models with Knowledge Graph Reasoning for Question Answering**

Research Questions

- How can reasoning within knowledge graphs be effectively integrated into language models to enhance their performance in Open-Domain Question Answering (ODQA)?

- How does multi-hop reasoning within knowledge graphs help language models answer complex questions?

- How can reasoning paths be generated alongside answers to enhance model explainability?

- Paper: **Empowering Language Models with Knowledge Graph Reasoning for Question Answering**

Problem Definition #1

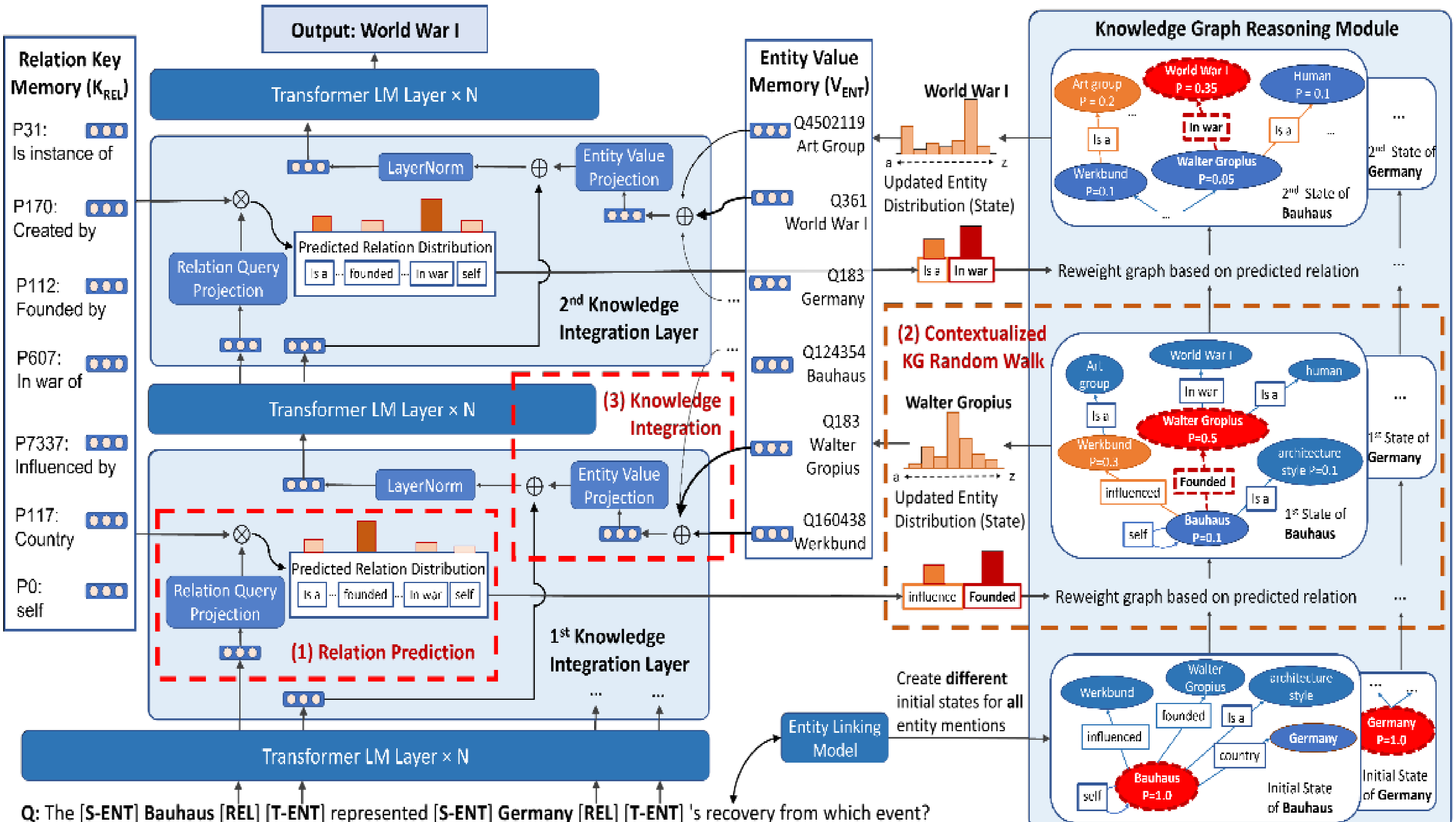
- **Knowledge Storage and Reasoning**: Limited explicit storage in LMs, difficulty answering complex, knowledge-rich questions.
- **Multi-Hop Reasoning**: Need for multiple steps to reach an answer; existing models often limited to single-step reasoning.
- **Explainability**: Transparency in reasoning paths is often lacking, affecting interpretability.

High level idea

The proposed OREOLM in the paper includes a key component—the Knowledge Interaction Layer (KIL). Its design approach is as follows:

- Interactive Reasoning Mechanism
- Multi-step Reasoning Path
- End-to-End Training

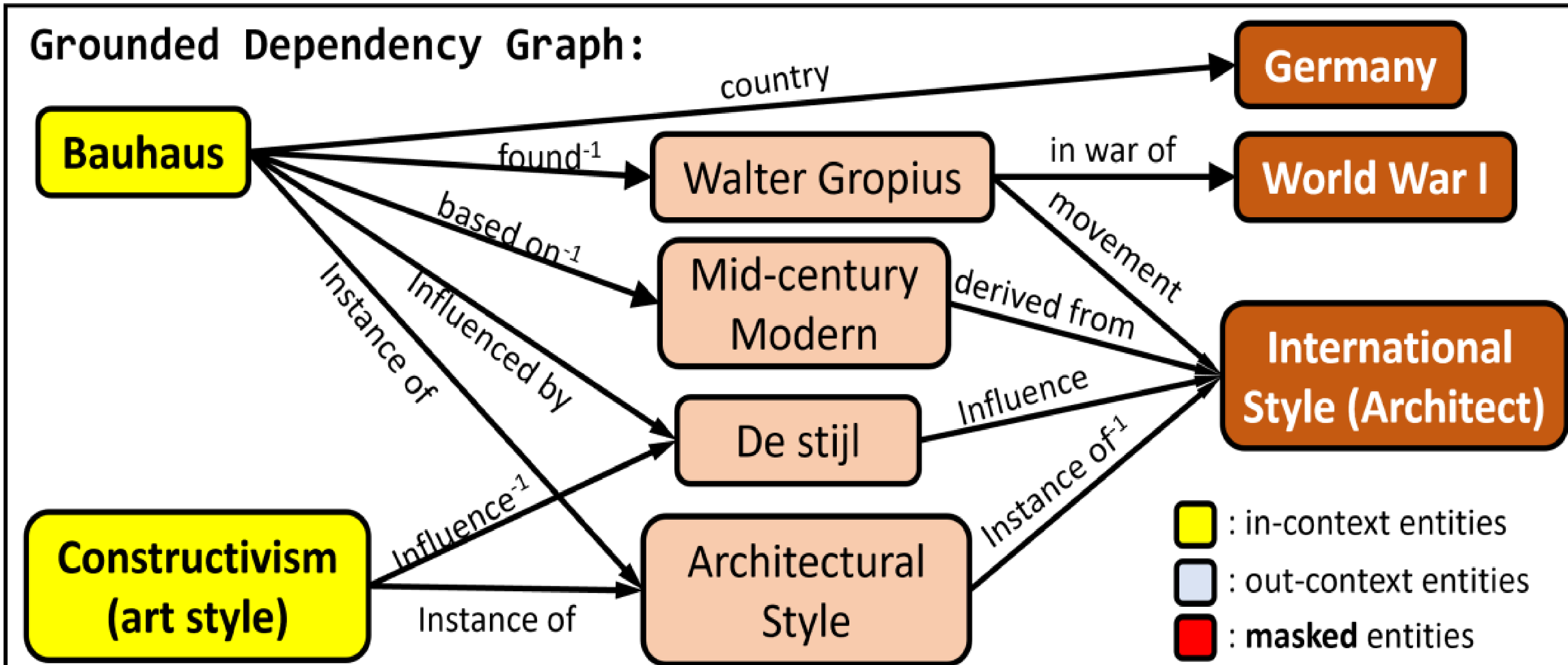
Algorithm #1



- Paper: **Empowering Language Models with Knowledge Graph Reasoning for Question Answering**

One example for pretraining model

Wiki-Passage: After [MASK]~~(Germany)~~'s defeat in [MASK]~~(world war I)~~ and…… influenced by the cultural experimentation such as **[S-ENT] constructivism [REL][T-ENT]**…… **[S-ENT] Bauhaus [REL][T-ENT]** also known as [MASK]~~(International Style)~~, was ……



Experiment Analysis-1

Models	#param	NQ	WQ	TQA	ComplexWQ	HotpotQA
T5 (Base)	0.22B	25.9	27.9	29.1	11.6	22.8
+ OREOLM (T=1)	0.23B + 0.68B	28.3	30.6	32.4	20.8	24.1
+ OREOLM (T=2)	0.24B + 0.68B	28.9	31.2	33.7	23.7	26.3
T5 (Large)	0.74B	28.5	30.6	35.9	16.7	25.3
+ OREOLM (T=1)	0.75B + 0.68B	30.6	32.8	39.1	24.5	28.2
+ OREOLM (T=2)	0.76B + 0.68B	**31.0**	**34.3**	**40.0**	**27.1**	**31.4**
T5-3B (Roberts et al., 2020)	3B	30.4	33.6	43.4	-	27.8
T5-11B (Roberts et al., 2020)	11B	32.6	37.2	50.1	-	30.2

Table 2: **Closed-Book Generative QA** performance of Encoder-Decoder LM on Single- and Multi-hop Dataset.

Models	#param (B)	WQ-SP	TQA
EaE (Févry et al., 2020)	0.11 + 0.26	62.4	24.4
FILM (Verga et al., 2021)	0.11 + 0.72	78.1	37.3
KEPLER (Wang et al., 2019)	0.12	48.3	24.1
RoBERTa (Base)	0.12	43.5	21.3
+ OREOLM (T=1)	0.12 + 0.68	80.1	39.7
+ OREOLM (T=2)	0.13 + 0.68	**80.9**	**40.3**
Ablation Studies			
RoBERTa + Concat KB + $\mathcal{L}_{SSM}$	0.12	47.1	22.6
+ OREOLM (T=2) w/o PT	0.13 + 0.68	46.9	22.7
w. $\mathcal{L}_{SSM}$	0.13 + 0.68	51.9	26.8
w. $\mathcal{L}_{SSM} + \mathcal{L}_{ent}$	0.13 + 0.68	68.4	35.7

Table 3: **Closed-Book Entity Prediction** performance of Encoder LM on WikiData-Answerable Dataset.

Experiment Analysis-2

Models	#param (B)	NQ	TQA
Graph-Retriever (Min et al., 2019)	0.11	34.7	55.8
REALM (Guu et al., 2020)	0.33 + 16	40.4	-
DPR (Karpukhin et al., 2020) + BERT	0.56 + 16	41.5	56.8
+ **OREOLM** (DPR, T=2)	0.57 + 17	43.7	58.5
FiD (Base) = DPR + T5 (Base)	0.44 + 16	48.2	65.0
+ **OREOLM** (T5, T=2)	0.45 + 17	49.3	67.1
+ **OREOLM** (DPR & T5, T=2)	0.46 + 17	51.1	68.4
FiD (Large) = DPR + T5 (Large)	0.99 + 16	51.4	67.6
+ **OREOLM** (T5, T=2)	0.99 + 17	52.4	68.9
+ **OREOLM** (DPR & T5, T=2)	1.00 + 17	**53.2**	69.5
KG-FiD (Base) (Yu et al., 2022a)	0.44 + 16	49.6	66.7
KG-FiD (Large) (Yu et al., 2022a)	0.99 + 16	**53.2**	69.8
EMDR^2 (Sachan et al., 2021b)	0.44 + 16	52.5	**71.4**

Table 4: **Open-Book QA** Evaluation.

Tests OREOLM's ability to reason about missing relationships

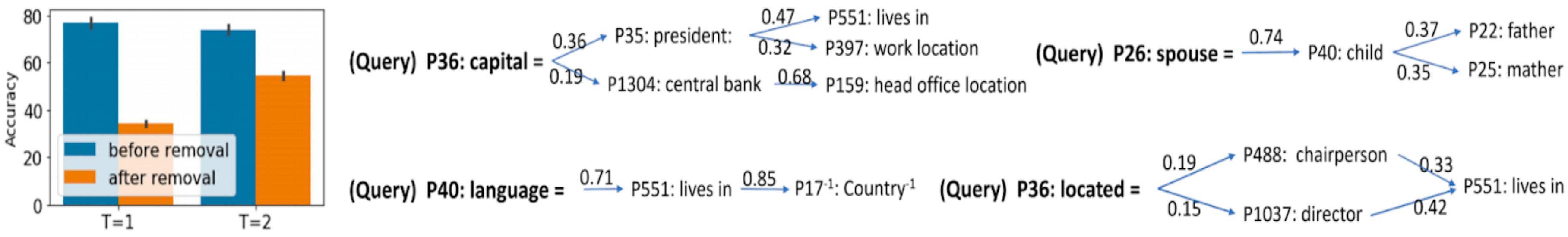


Figure 4: **Testing the reasoning capacity of OREOLM to infer missing relations**. On the **left**, the barplot shows the transfer performance on EQ before and after removing relation edges, OREOLM ($T = 2$) is less influenced. On the **right** shows reasoning paths (rules) automatically generated by OREOLM for each missing relation.

Roadmap

- Background and Motivation
  - Background
  - Research questions
- Algorithms
  - Algorithm #1
  - Algorithm #2
  - Algorithm #3
  - Query2Box
- Conclusion

Query2Box: Pro/Con

Pro:

- Embeds complex queries efficiently with logical conjunctions and disjunctions and existing quantifiers efficiently
- Improves multi-hop reasoning for incomplete graphs, showing good results for generalizing across unseen queries

Con:

- Computational burden for training and Disjunctive Normal Form transformation
- Higher embedding dimension needed to maintain accuracy

BetaE: Pro/Con

Pro:

- Beta Embedding can naturally model uncertainty by its differential entropy
- Beta Embedding firstly support full first-order logic by extenting the negation operation
- Beta Embedding provides a closure definition for operators and allows the operators to combine in arbitrary ways at a fixed space/time complexity

Con:

- The use of De Morgan’s laws for union operations is an approximation, which lead to reduced accuracy for certain queries requiring precise union handling.
- BetaE is designed for static KGs which need extra work to apply on dynamically changing knowledge graphs

NewLook: Pro/Con

- Pro:
  - General Applicability: NewLook supports more variables than previous work
  - Effectiveness: NewLook goes beyond the linear transformation, thus performs better than previous work
  - Efficency: NewLook runs on average 3 times faster
- Con:
  - Long training time, which suggests high computational costs or complexity at starting condition
  - Does not incorporate subgraph matching and embedding based methods

Pros:

- **Enhanced Reasoning Ability: OREOLM can infer missing relations, enabling it to answer questions that require multi-hop reasoning across a knowledge graph.**
- **Interpretability: By providing explicit reasoning paths, OREOLM improves the transparency of the answer generation process.**
- **Improved QA Performance: It demonstrates significant performance improvements in both closed-book and open-book QA across different datasets.**

Cons:

- **Complexity: Integrating KG reasoning adds architectural and training complexity, which may lead to higher computational requirements.**
- **Dependency on Knowledge Graphs: The model’s effectiveness relies heavily on the quality and completeness of the knowledge graph; missing or incorrect relations can impair its reasoning.**
- **Scalability: Very large knowledge graphs may introduce challenges with memory and processing power as the number of entities and relationships scales up.**