COMBINING REPRESENTATION LEARNING AND LOGICAL RULE REASONING FOR KNOWLEDGE GRAPH INFERENCE

Yizhou Sun

Department of Computer Science

University of California, Los Angeles

yzsun@cs.ucla.edu

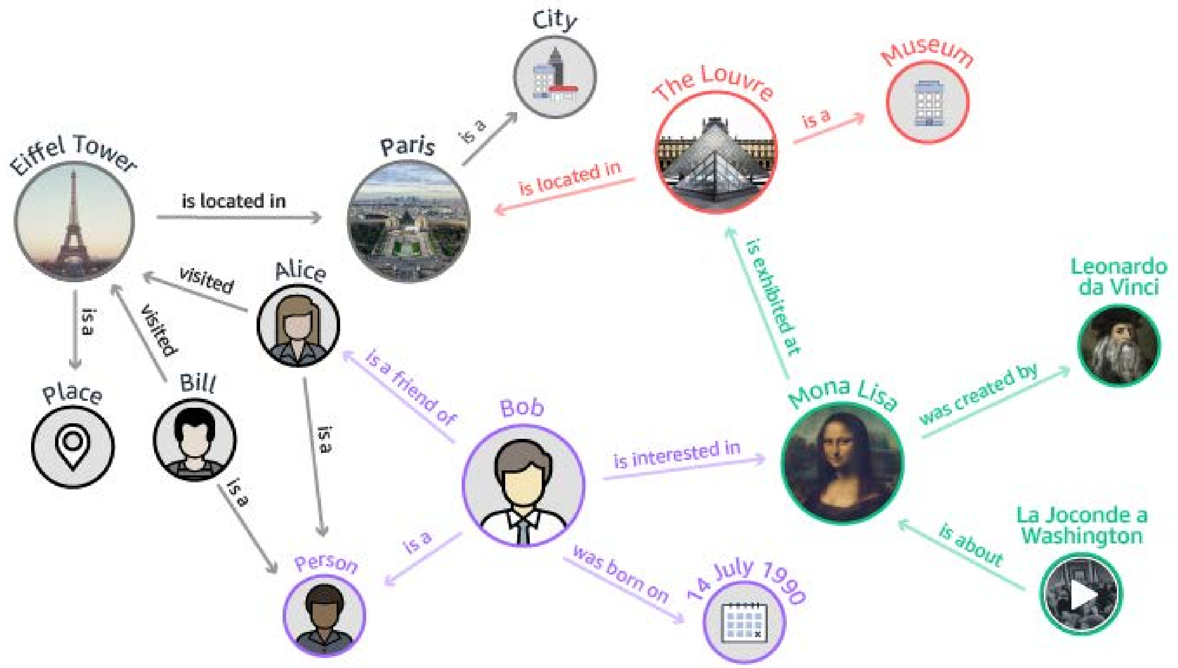
August 22, 2022

Outline

- Introduction
- UniKER: Integrating Logical Rule into KGE
- FuzzQE: KGE based Fuzzy Logic for Logical Query
- Summary

Knowledge Graph

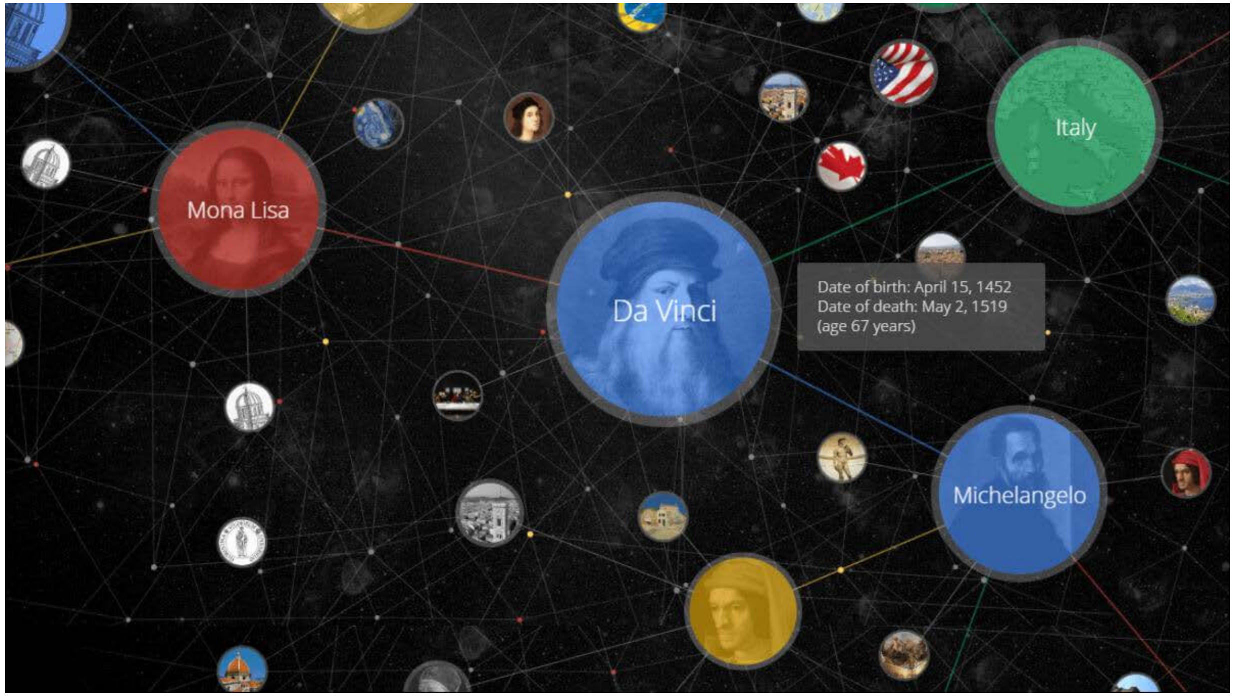
- What are knowledge graphs?
  - Multi-relational graph data
    - (heterogeneous information network)
  - Provide structured representation for semantic relationships between real-world entities



A triple (h, r, t) represents a fact, ex:
(Eiffel Tower, is located in, Paris)

Knowledge Graph Embedding

- Entities: low dimensional vectors
- Relations: parametric algebraic operators
- Triples: representation-based score function



Summary of Existing Approaches

- Define a score function for a triple: $f_r(\boldsymbol{h}, \boldsymbol{t})$
  - According to entity and relation representation
- Define a loss function to guide the training
  - E.g., an observed triple scores higher than a negative one

Model	Score Function	
SE (Bordes et al., 2011)	$-\lVert \boldsymbol{W}_{r,1}\mathbf{h} - \boldsymbol{W}_{r,2}\mathbf{t} \rVert$	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k, \boldsymbol{W}_{r,\cdot} \in \mathbb{R}^{k \times k}$
TransE (Bordes et al., 2013)	$-\lVert \mathbf{h} + \mathbf{r} - \mathbf{t} \rVert$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
TransX	$-\lVert g_{r,1}(\mathbf{h}) + \mathbf{r} - g_{r,2}(\mathbf{t}) \rVert$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
DistMult (Yang et al., 2014)	$\langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}(\langle \mathbf{r}, \mathbf{h}, \overline{\mathbf{t}} \rangle)$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$
HolE (Nickel et al., 2016)	$\langle \mathbf{r}, \mathbf{h} \otimes \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ConvE (Dettmers et al., 2017)	$\langle \sigma(\mathrm{vec}(\sigma([\overline{\mathbf{r}}, \overline{\mathbf{h}}] * \boldsymbol{\Omega}))\boldsymbol{W}), \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
RotatE	$-\lVert \mathbf{h} \circ \mathbf{r} - \mathbf{t} \rVert^2$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k, \lvert r_i \rvert = 1$

Source: Sun et al., RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space (ICLR'19)

Knowledge Graph Inference

- Knowledge Graph Completion
  - Given an incomplete triple, infer the missing entity
  - E.g., **(Eiffel Tower, is located in, ?)**
- Logical Query
  - Given a more complicated query, infer the entity
  - E.g.,

$$q = V_?: \exists V \quad (Compose(\text{John Lennon}, V) \lor Compose(\text{Paul McCartney}, V)) \land \neg AwardedTo(\text{Grammy Award}, V) \land SungBy(V, V_?)$$

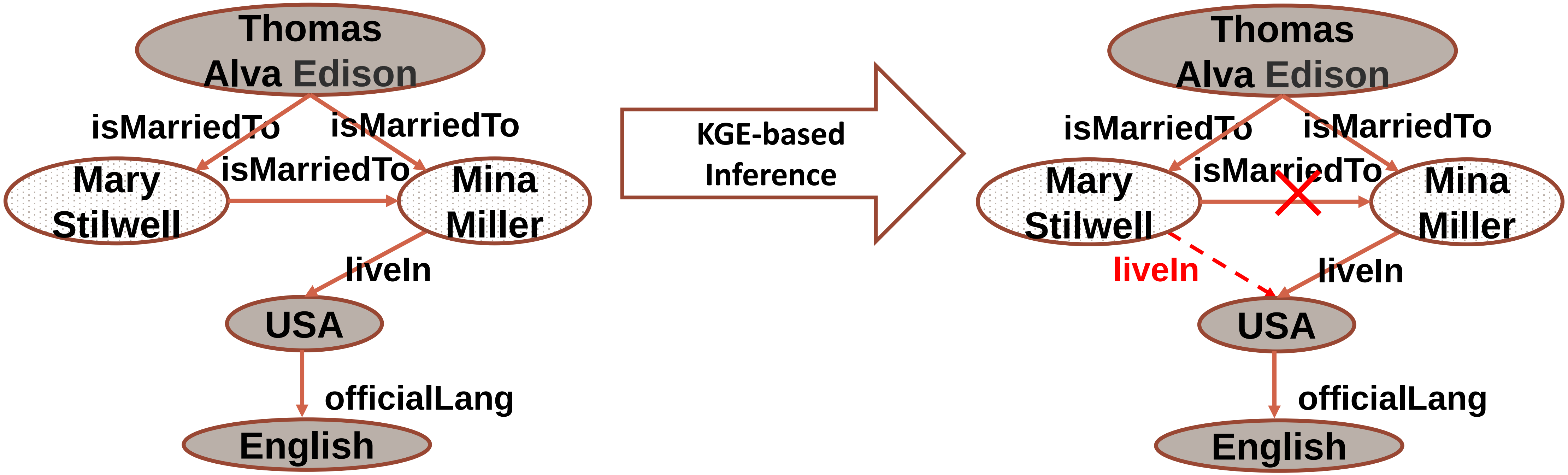
    - Return singers that have sung songs written by Lennon or McCartney but never won Grammy Award

Outline

- Introduction
- UniKER: Integrating Logical Rule into KGE (EMNLP'21)
  - By Kewei Cheng et al.
- FuzzQE: KGE based Fuzzy Logic for Logical Query
- Summary

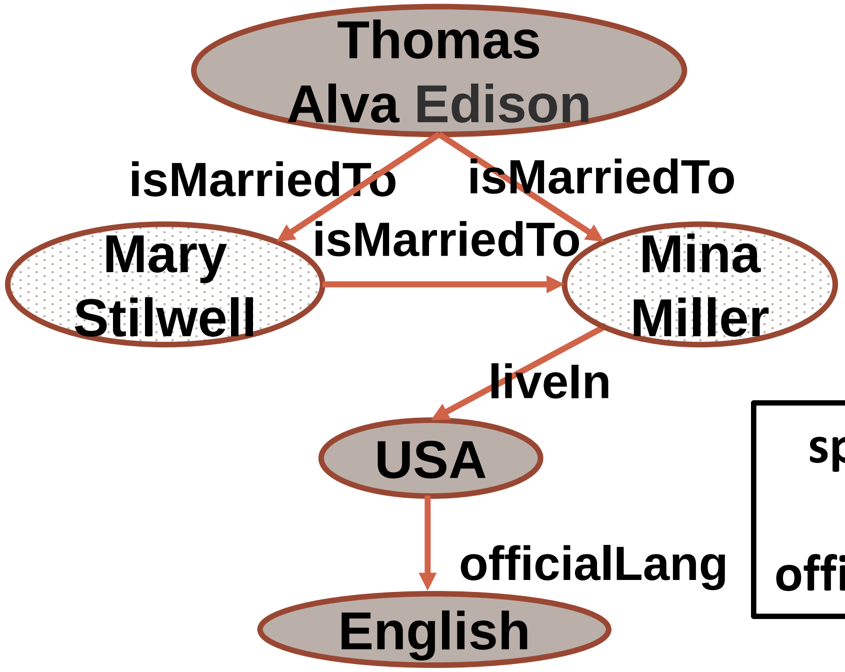
Knowledge Graph Embedding-based KG reasoning

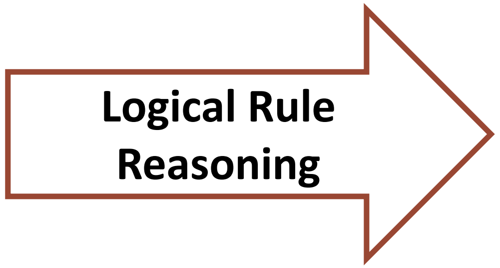
- Pros: Shows good scalability as well as robustness.
- Cons: Fails to capture high-order dependency between entities and relations.

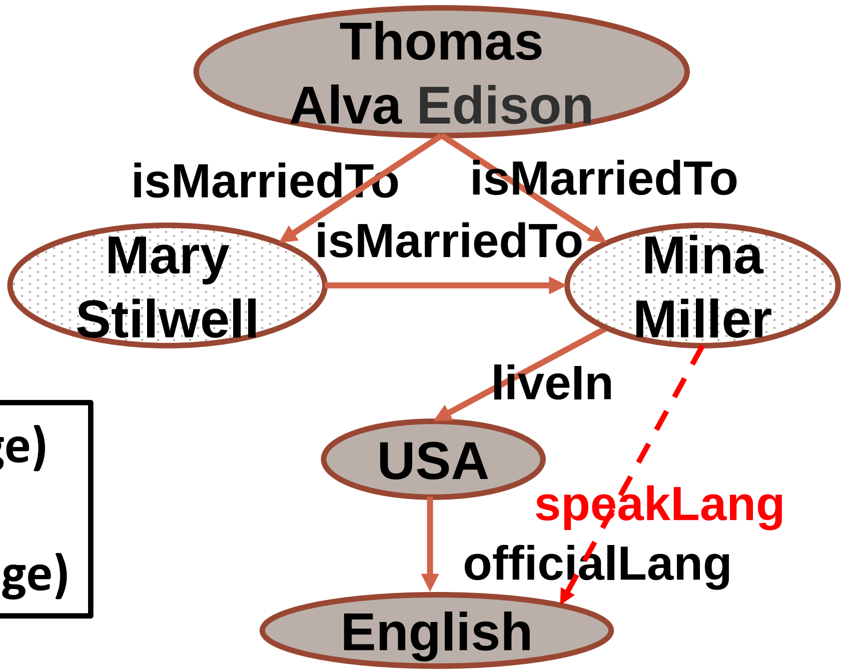


Logical Rule-based KG reasoning

- Pros: good at capturing high-order dependency.
- Cons: unable to handle noisy data as well as suffer from high computation complexity.



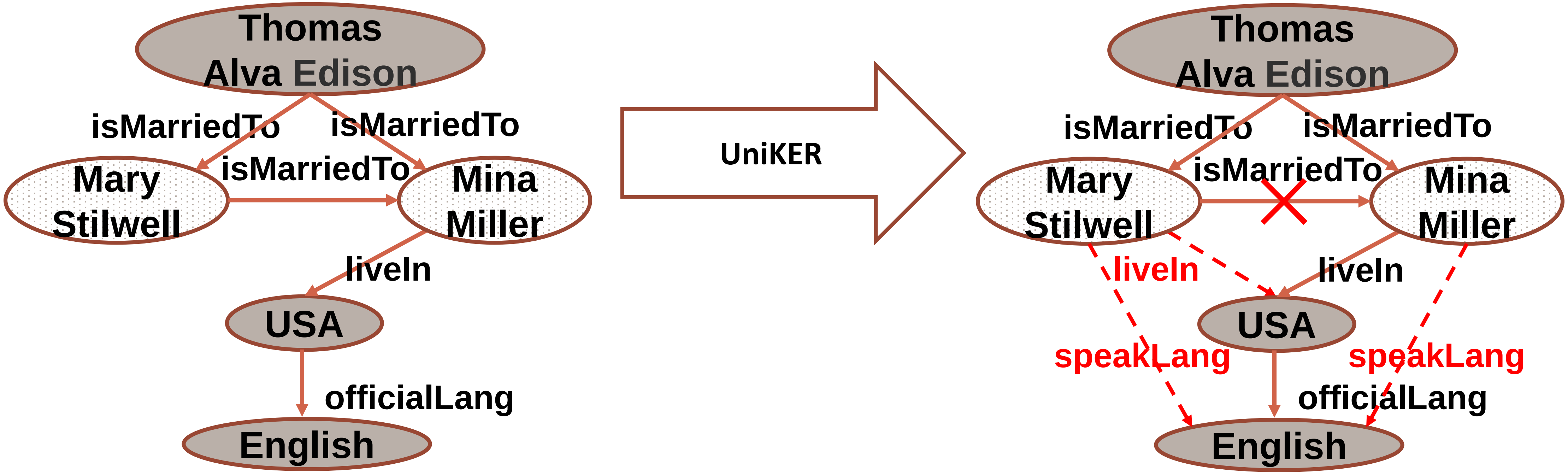




speakLanguage(Person, Language)
⇐ liveIn(Person, Country) ∧
officialLanguage (Country, Language)

Definite Horn Rules

Enriched by both:1+1>2!



Combining Both Worlds

- Connecting the two worlds

Knowledge Graph	Logic	Example
Entities	Constant	Miller
Relation	Predicate	liveIn(x, y)
Triple (a link on KG)	Ground predicate	liveIn(Miller, USA)
A Path on KG	A conjunction of ground predicates	liveIn(Thomas Alva Edison, USA) ∧ officialLanguage (USA, English)

Existing Work

- Probabilistic logic is widely used to integrate both worlds
  - PSL-based Regularization in Embedding Loss
    - Leverage Probabilistic Soft Logic (PSL) [7] for satisfaction loss calculation
    - Treat logical rules as additional regularization to embedding models, where the satisfaction loss of ground rules is integrated into the original embedding loss.
    - Limitation: only utilize a sample set of rule instances
  - Embedding-based Variational Inference for MLN.
    - Extends Markov Logic Network (MLN) [8]
    - Leverage graph embedding to define variational distribution for all possible hidden triples to conduct variational inference of MLN.
    - Limitation: efficiency issue, sampling is required

Limitations of Existing Work

Categories	Methods	Interactive	Exact Logical Inference
PSL-based Regularization	KALE [1]	×	×
	RUGE [2]	√	×
	Rocktaschel et al [3]	×	×
Embedding-based Variational Inference to MLN	pLogicNet [4]	√	×
	ExpressGNN [5]	√	×
	pGAT [6]	√	×

Our Proposed Work: UniKER

- Idea 1: logical reasoning => enhance KG => enhance embedding
  - use forward chaining to conduct exact inference
- Idea 2: Embedding => enhance KG => enhance logical reasoning
  - Adding potentially useful triples
  - Removing potentially incorrect triples
- Idea 3: combine embedding and logical rules in an iterative manner.

Traditional Logical Inference: MAX-SAT problem

Knowledge Graph

Entities

Country
USA

Person
Thomas Alva Edison, Mary Stilwell, Mina Miller

Language
English

Predicates

isMarriedTo
liveIn
isParentOf
isSiblingOf
officialLanguage
speakLanguage

Observed Facts

isMarriedTo (Thomas Alva Edison, Mary Stilwell)
isMarriedTo (Thomas Alva Edison, Mina Miller)
isMarriedTo (Mary Stilwell, Mina Miller)
liveIn (Mina Miller, USA)
officialLanguage (USA, English)

⇩

All ground predicates

liveIn (Thomas Alva Edison, USA)	**T**
…	
liveIn (Mary Stilwell , USA)	**?**

speakLanguage(Person, Language) ⇐ liveIn(Person, Country) ∧ officialLanguage (Country, Language)

Definite Horn rule

All ground rules

speakLanguage (Thomas Alva Edison, English) ⇐ liveIn (Thomas Alva Edison, USA) ∧ officialLanguage (USA, English)

…

speakLanguage (Mary Stilwell, English) ⇐ liveIn (Mary Stilwell, USA) ∧ officialLanguage (USA, English)

NP-complete ⇩ **SAT Solver**

New fact: speakLanguage (Mina Miller, English)

Forward Chaining for Horn rules: Exact and Fast

Observed Facts

isMarriedTo (Thomas Alva Edison, Mary Stilwell)
isMarriedTo (Thomas Alva Edison, Mina Miller)
isMarriedTo (Mary Stilwell, Mina Miller)
liveIn (Mina Miller, USA)
officialLanguage (USA, English)

Country = USA
Language = English

Person = Mina Miller

speakLanguage(Person, Language) ⇐ liveIn(Person, Country) ∧ officialLanguage (Country, Language)

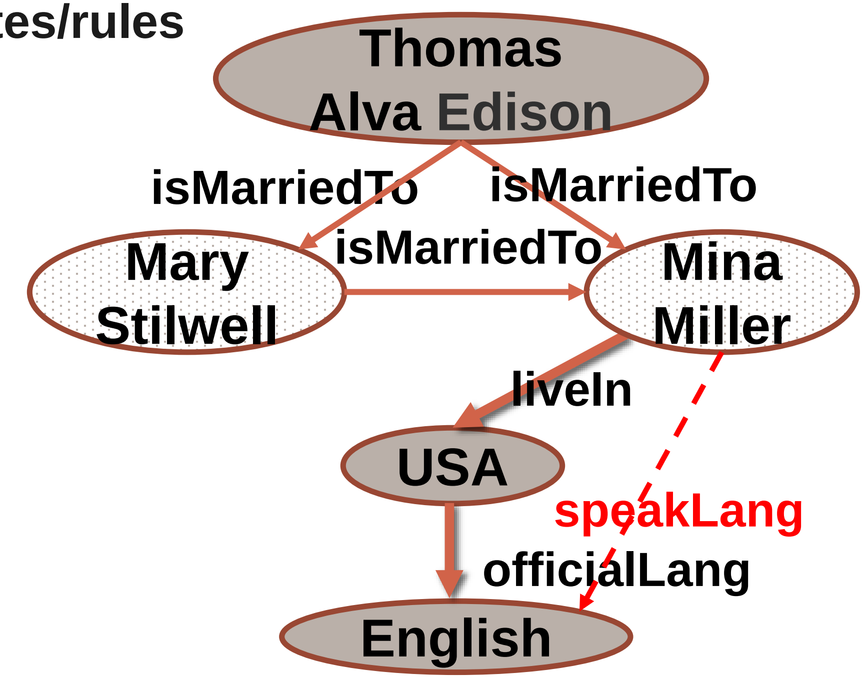
Definite Horn rule

Forward Chaining

speakLanguage (Mina Miller, English)

New fact

involve only a small subset of active ground predicates/rules



Path Traverse

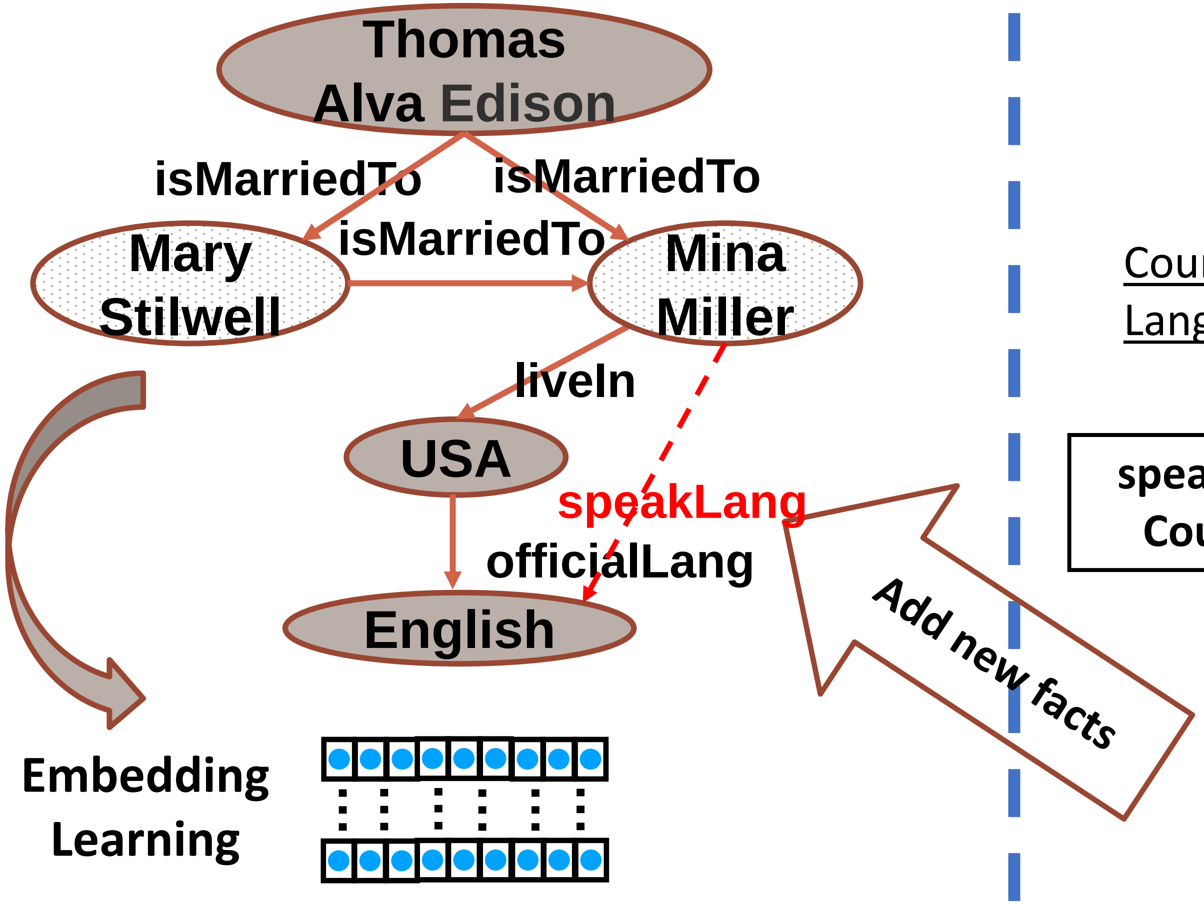
Iterative Mutual Enhancement

- Enhance KGE via logical inference
  - Update KG via forward chaining-based logical reasoning

- Enhance logical inference via KGE
  - Excluding potentially incorrect triples
  - Including potentially useful hidden triples

Update KG via Forward Chaining-based Logical Reasoning

Knowledge Graph Embedding



Logical Rule-based Reasoning

liveIn (Mina Miller, USA)
officialLanguage (USA, English)

Observed Facts

Country = USA
Language = English

speakLanguage(Person, Language) ⇐ liveIn(Person, Country) ∧ officialLanguage (Country, Language)

Definite Horn rule

Logical Inference

speakLanguage (Mina Miller, English)

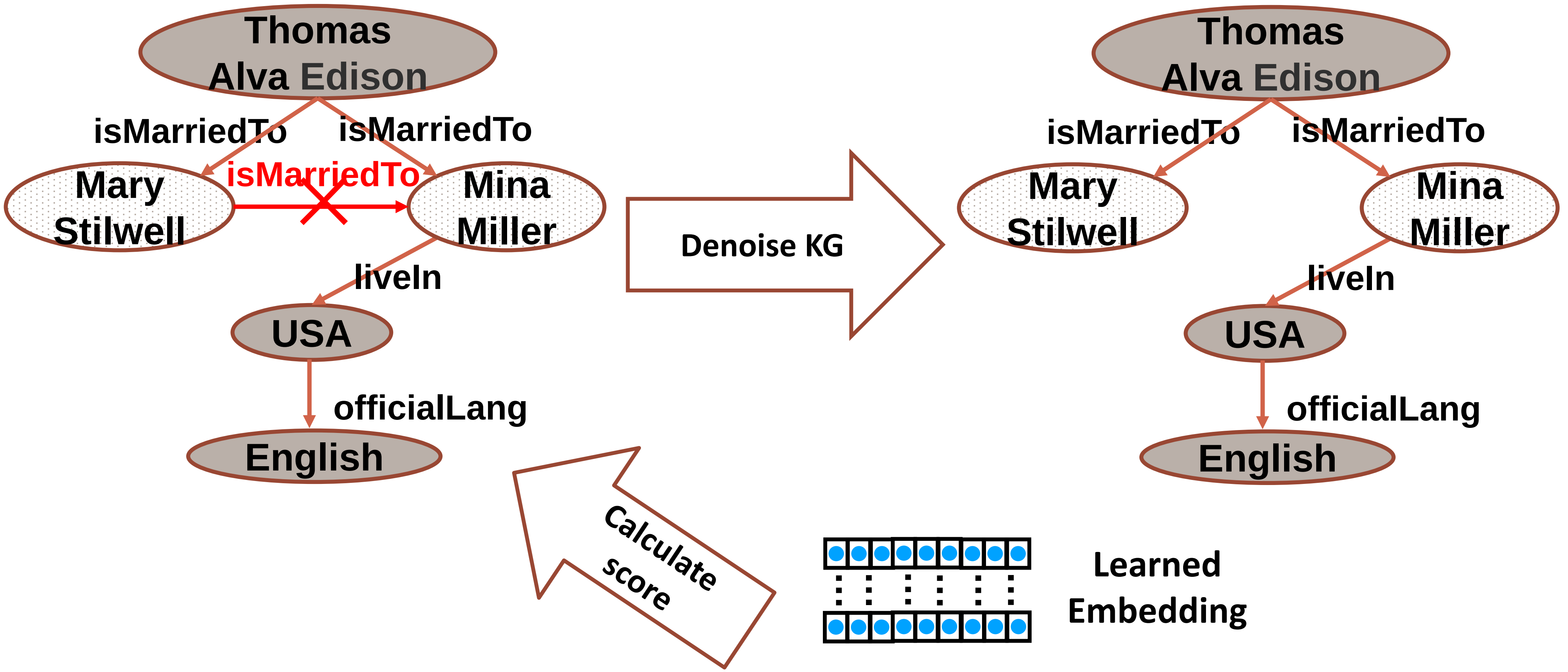
New fact

Iterative Mutual Enhancement

- Enhance KGE via logical inference
  - Update KG via forward chaining-based logical reasoning

- Enhance logical inference via KGE
  - Excluding potential incorrect triples
  - Including potential useful hidden triples

Excluding potential incorrect triples



Including potential useful hidden triples

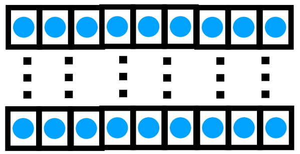
√ triples in KGs
? triples not in KGs

Observed Facts

liveIn (Mary Stilwell, USA) **?**
officialLanguage (USA, English) **√**

Exist?

Learned Embedding



Country = USA
Language = English

Person = Mary Stilwell

speakLanguage(Person, Language) ⇐ liveIn(Person, Country) ∧ officialLanguage (Country, Language)

Definite Horn rule

Forward Chaining

Including potential useful hidden triples

√ triples in KGs
? triples not in KGs

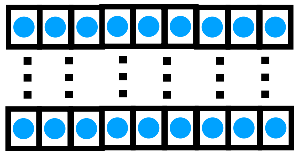
Observed Facts

liveIn (Mary Stilwell, USA) **√**
officialLanguage (USA, English) **√**

Exist?

Add!

Learned Embedding



Country = USA
Language = English

Person = Mary Stilwell

speakLanguage(Person, Language) ⇐ liveIn(Person, Country) ∧ officialLanguage (Country, Language)

Definite Horn rule

Forward Chaining

speakLanguage (Mina Miller, English)

New fact

Experimental Results

- KG completion task

Model	Kinship Hit@1	Kinship Hit@10	Kinship MRR	FB15k-237 Hit@1	FB15k-237 Hit@10	FB15k-237 MRR	WN18RR Hit@1	WN18RR Hit@10	WN18RR MRR
RESCAL	0.489	0.894	0.639	0.108	0.322	0.179	0.123	0.239	0.162
SimplE	0.335	0.888	0.528	0.150	0.443	0.249	0.290	0.351	0.311
HypER†	0.364	0.903	0.551	0.252	0.520	0.341	0.436	0.522	0.465
TuckER†	0.373	0.898	0.567	0.266	0.544	0.358	**0.443**	0.526	0.470
BLP†	-	-	-	0.062	0.150	0.092	0.187	0.358	0.254
MLN	0.655	0.732	0.694	0.067	0.160	0.098	0.191	0.361	0.259
KALE	0.433	0.869	0.598	0.131	0.424	0.230	0.032	0.353	0.172
RUGE	0.495	0.962	0.677	0.098	0.376	0.191	0.251	0.327	0.280
ExpressGNN	0.105	0.282	0.164	0.150	0.317	0.207	0.036	0.093	0.054
pLogicNet	0.683	0.874	0.768	0.261	0.567	0.364	0.301	0.410	0.340
pGAT†	-	-	-	0.377	0.609	0.457	0.395	0.578	0.459
BoxE†	-	-	-	-	0.538	0.337	-	0.541	0.451
TransE	0.221	0.874	0.453	0.231	0.527	0.330	0.007	0.406	0.165
UniKER-TransE	0.873	**0.971**	0.916	0.463	**0.630**	0.522	0.040	0.561	0.307
DistMult	0.360	0.885	0.543	0.220	0.486	0.308	0.304	0.409	0.338
UniKER-DistMult	0.770	0.945	0.823	**0.507**	0.587	0.533	0.432	0.538	0.485
RotatE	0.787	0.933	0.862	0.237	0.526	0.334	0.421	0.563	0.469
UniKER-RotatE	**0.886**	**0.971**	**0.924**	0.495	0.612	**0.539**	0.437	**0.580**	**0.492**

Experimental Results

- A few iterations is good enough

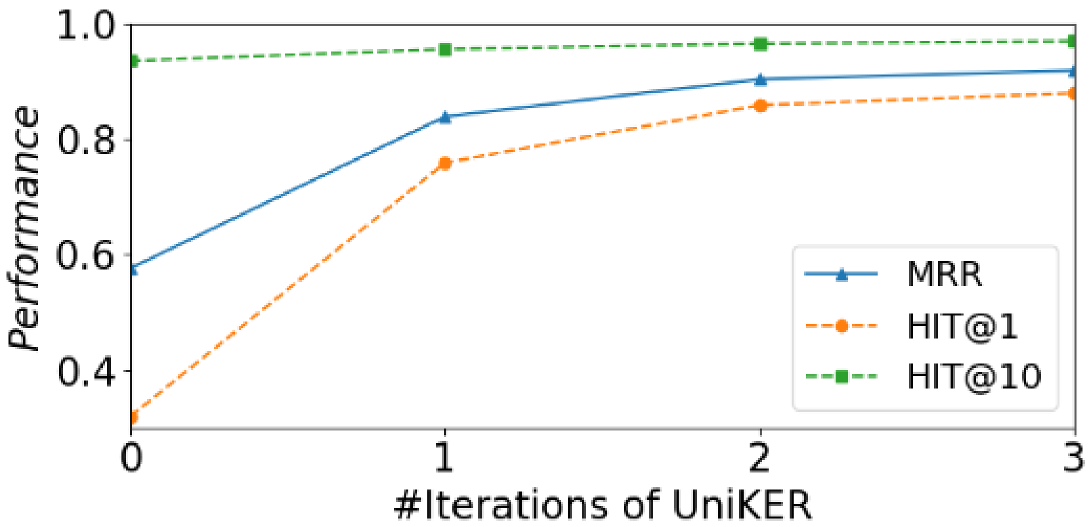


Figure 3: Impact of #iterations on UniKER (KG completion task on Kinship dataset).

Robust to Noise

- construct a noisy dataset with noisy triples to be 40% of original data.

Model	θ	Hit@1	Hit@10	MRR
TransE	-	0.026	0.800	0.319
UniKER-TransE	10	0.286	0.776	0.466
	20	0.311	0.816	0.503
	30	0.322	**0.833**	0.520
	40	**0.352**	0.812	**0.523**
	50	0.292	0.791	0.486

Table 3: Ablation study on noise threshold $\theta\%$ on Kinship dataset (whose train set is injected with noise)

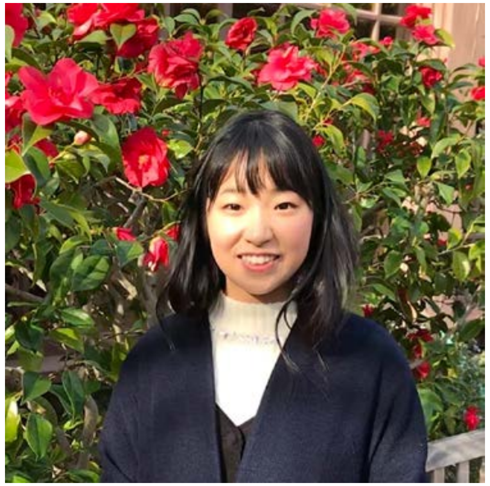
Efficient

- Evaluate the scalability of forward chaining against a number of SOTA inference algorithms for MLN

Model	sub-YAGO3-10	sub-Kinship	RC1000	Kinship	FB15k-237	WN18RR
MCMC	76433s	-	-	-	-	-
MCSAT	1292s	25912s	-	-	-	-
BP	10s	16343s	-	-	-	-
liftedBP	15s	16075s	-	-	-	-
Tuffy	0.849s	1.398s	4.899s	-	-	-
Forward Chaining	0.003s	0.034s	0.007s	0.593s	186s	30s

Table 7: Comparison of Inference Time for Forward Chaining vs. MLN.

Outline



- Introduction
- UniKER: Integrating Logical Rule into KGE
- KGE based Fuzzy Logic for Logical Query (AAAI'22)
  - By Xuelu Chen and Ziniu Hu et al.
- Summary

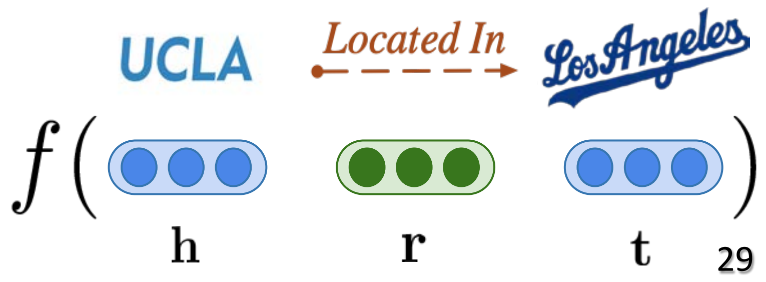
From Link Prediction to Multi-Hop Logical Reasoning

Link Prediction

$(UCLA, LocatedIn, ?)$

Triple

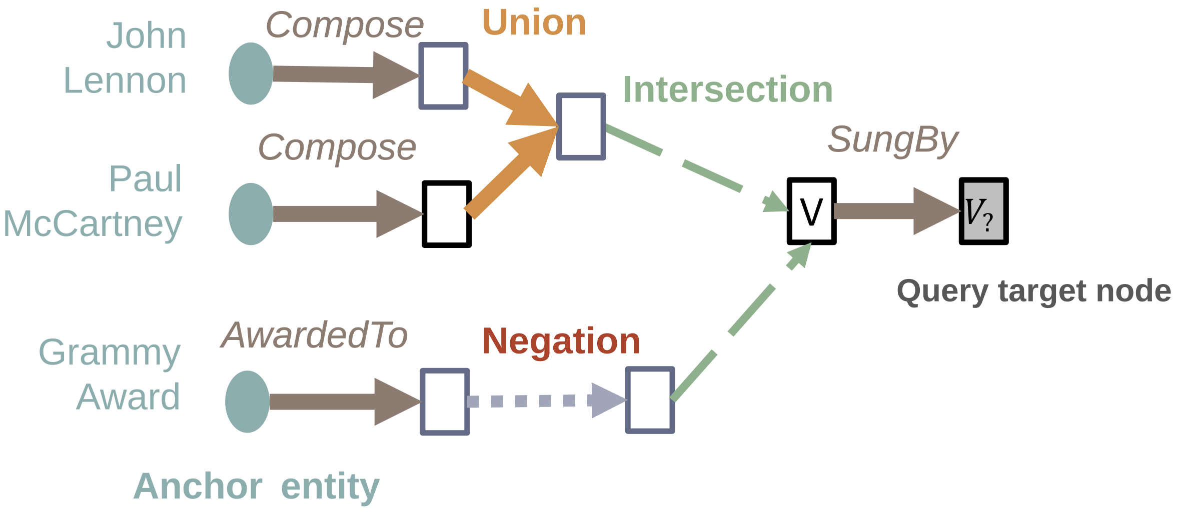
Score Function



Can we handle more complex queries on KGs?

First-Order Logic (FOL) Queries

$$q = V_?: \exists\, V \quad (Compose(\text{John Lennon}, V) \lor Compose(\text{Paul McCartney}, V)) \land \neg AwardedTo(\text{Grammy Award}, V) \land SungBy(V, V_?)$$



Reasoning on Knowledge Graphs

- Methods (1)
  - Traverse the KG to search for results
    - e.g. by subgraph matching (Gstore [Zou, VLDB'2011])
    - Drawbacks:
      - **Incompleteness of KGs**
        - Real-world KGs are often severely incomplete
        - A single missing edge may make the query unanswerable
          - Impossible to get answers for many queries by directly traversing KG
      - **Computation Complexity**
        - Wikidata reports that their query engine performance falls off a cliff and may time out, when the number in a group of interest (e.g. people born in France) exceeds a certain threshold

How can we make it faster and make it robust to missing edges?

Reasoning on Knowledge Graphs

- Methods (2)
  - Logical query embedding models
    - Embed logical queries and entities in the same vector space and conduct query answering via dense similarity search.
    - Representative works:
      - GQE [Hamilton et al., NeurIPS'2018], Query2Box [Ren et al., ICLR'2020], BetaE [Ren & Leskovec, NeurIPS'2020], etc.
  - Merits
    - Can handle missing edges
    - No need to model intermediate entities
      - Inference in constant time with locality sensitive hashing

Reasoning on Knowledge Graphs

- Methods (2)
  - Logical query embedding models
    - Challenges
      - These logical operators are parameterized so that they require a large number of complex FOL queries as training data
        - Greatly limits the scope of application
          - Such data is often arduous or even inaccessible to collect in most real-world KGs!!
      - Does not satisfy the axiomatic systems of classical logic
        - Limits inference accuracy

Our solution: FuzzQE

- Merits
  - FuzzQE satisfies the axioms of logical operations and is capable of preserving the logical operation properties in vector space
    - Significantly better performance to the state-of-the-art methods in answering FOL queries.
  - Logical operators do not require learning any operator specific parameters
    - Even it is trained with only link prediction and no complex queries, it works well
      - Comparable to state-of-the-art models that are trained with extra complex query data
      - Significantly outperforms previous models under the same training condition (link prediction only)

Challenging Questions

- Combining Representation Learning with Logical Reasoning
  - How to represent an entity?
    - Point? Box? Distribution?
  - How to represent a set from a subquery?
    - Point? Box? Distribution?
  - How to define an embedding-based function denoting an entity belonging to a set?
  - How to recursively define embedding for each logical expression?
  - How to define an embedding-based function for each logical operator (and, or, negation)?
  - How to preserve logical laws (additional constrains) that logical operators have to preserve?
    - Commutative, associative, etc.
  - How to train the model in a self-supervised way? (No additional Query-Answer pair)

Outline

- Bridging set and logical expressions
- Representing set, element, and membership
- Defining set operations that preserve logical laws
- Self-supervised training

Bridging set and logical expressions

- A FOL query corresponds to an answer set

$$q = V_?: \exists V \quad (Compose(\text{John Lennon}, V) \lor Compose(\text{Paul McCartney}, V)) \land \neg AwardedTo(\text{Grammy Award}, V) \land SungBy(V, V_?)$$

$$s(x) := \exists y \quad (Compose(\text{John Lennon}, y) \lor Compose(\text{Paul McCartney}, y)) \land \neg AwardedTo(\text{Grammy Award}, y) \land SungBy(y, x)$$

$$q \coloneqq \{x | s(x) \; is \; true\}$$

Logical operators vs. set operators

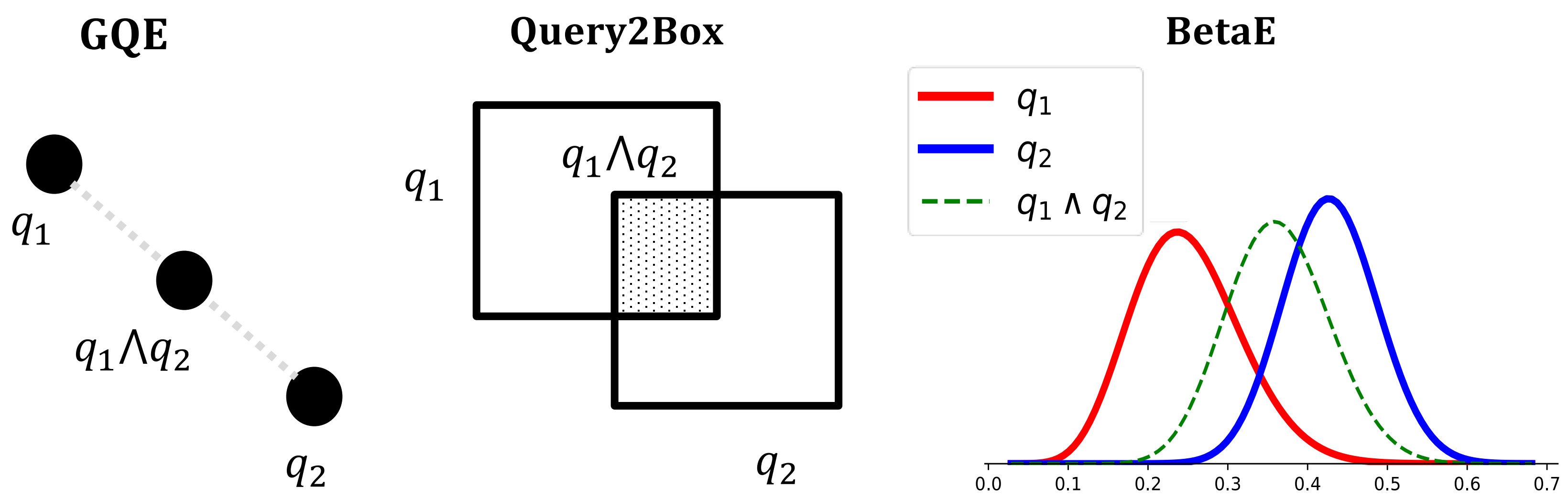
- Query Conjunction – Set Intersection
- Query Disjunction – Set Union
- Query Negation – Set Complement

Outline

- Bridging set and logical expressions
- Representing set, element, and membership
- Defining set operations that preserve logical laws
- Self-supervised training

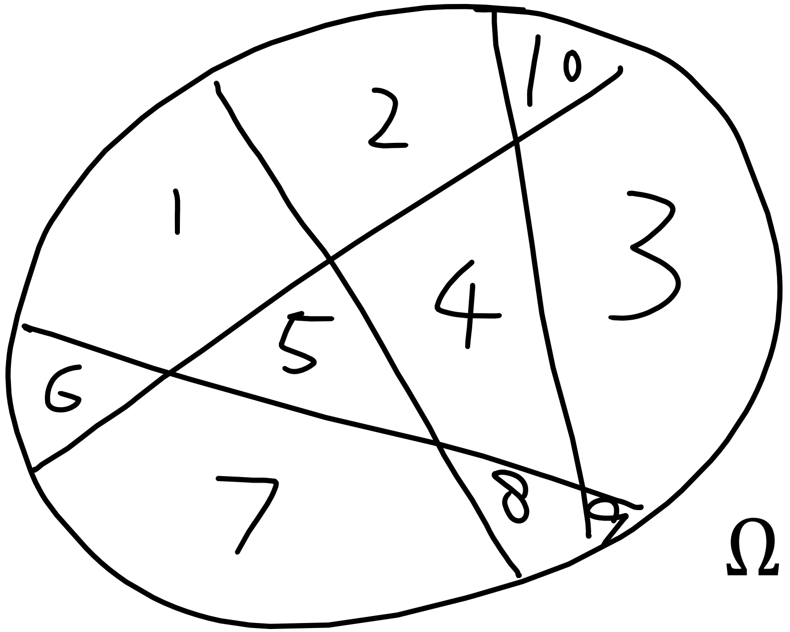
Representing set, element, and membership

- Existing approaches
  - $q, e, p(q|e)$?



Our approach

- Query as a fuzzy Set, which is represented by
  - $\boldsymbol{S}_q \in [0,1]^d$
  - Properties
    - $(1, \dots, 1): \Omega$
    - $(0, \dots, 0): \emptyset$
    - Subset, negation



- Entity as a stochastic vector
  - $\boldsymbol{P}_e \in [0,1]^d, and\ \sum_i \boldsymbol{P}_e(i) = 1$
- Embedding-based membership function
  - $$\phi(q, e) = \mathbb{E}_{e \sim \mathbf{p}_e}[e \in S_q] = \sum_{i=1}^{d} \Pr(e \in U_i) \Pr(U_i \subseteq S_q) = \mathbf{S}_q{}^{\intercal} \mathbf{p}_e$$

Outline

- Bridging set and logical expressions
- Representing set, element, and membership
- Defining set operations that preserve logical laws
- Self-supervised training

Defining set operations that preserve logical laws

- Representing atomic query (project an entity to a set)
  - $\mathbf{S}_q = \mathcal{P}_r(\mathbf{p}_e) = \mathbf{g}(\mathrm{LN}(\mathbf{W}_r \mathbf{p}_e + \mathbf{b}_r))$ e.g., *Compose*(John Lennon, V)
- Given the representations of two subqueries, define set operators via product logic (a special case of fuzzy logic)

$$q_1 \wedge q_2: \quad \mathcal{C}(\mathbf{S}_{q1}, \mathbf{S}_{q2}) = \mathbf{S}_{q1} \circ \mathbf{S}_{q2}$$

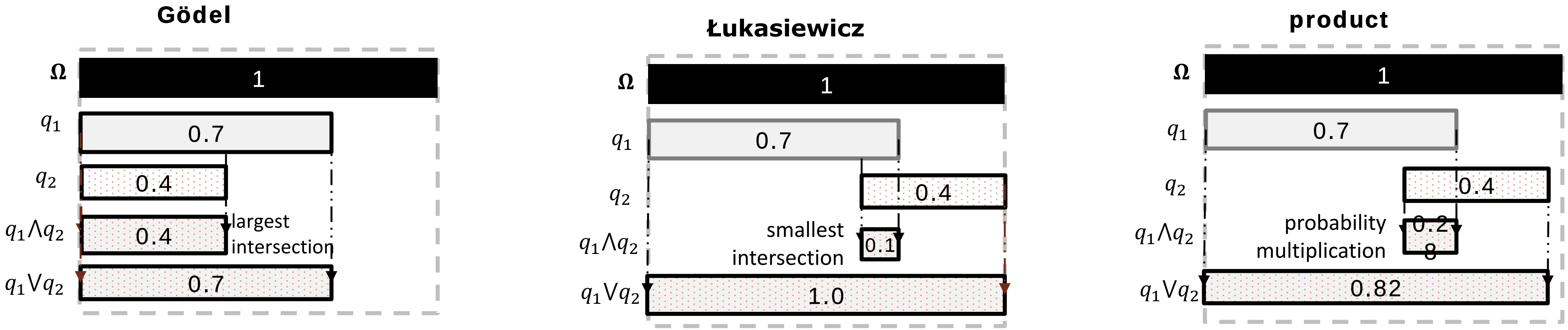
$$q_1 \vee q_2: \quad \mathcal{D}(\mathbf{S}_{q1}, \mathbf{S}_{q2}) = \mathbf{S}_{q1} + \mathbf{S}_{q2} - \mathbf{S}_{q1} \circ \mathbf{S}_{q2}$$

$$\neg q: \quad \mathcal{N}(\mathbf{S}_q) = 1 - \mathbf{S}_q$$

More about fuzzy logic

- negation: $c(x) = 1 - x$
- t-norm: conjunction (set intersection)
- t-conorm: disjunction (set union)
  - Defined by De Morgan's law

	t-norm ($\wedge$)	t-conorm ($\vee$)	Special Properties
minimum (Gödel)	$t(a,b) = \min(a,b)$	$s(a,b) = \max(a,b)$	idempotent
product	$t(a,b) = ab$	$s(a,b) = a + b - ab$	strict monotonicity
Łukasiewicz	$t(a,b) = \max(a+b-1, 0)$	$s(a,b) = \min(a+b, 1)$	nilpotent



Well, why are they good?

- **Consistent with logical axioms!**
  - An example of conjunction associativity

release(**the Beatles**, ?) ∧ compose(**John Lennon**, ?) ∧ compose(**Paul McCartney**, ?)
≡ release(**the Beatles**, ?) ∧ (compose(**John Lennon**, ?) ∧ compose(**Paul McCartney**, ?))

  - Previous approach GQE uses average as logical operator conjunction
    - **But,** $\frac{\frac{a+b}{2}+c}{2} \neq \frac{a+\frac{b+c}{2}}{2}$

$\boldsymbol{And}$: $(\boldsymbol{S_1} \circ \boldsymbol{S_2}) \circ \boldsymbol{S_3} = \boldsymbol{S_1} \circ (\boldsymbol{S_2} \circ \boldsymbol{S_3})$

Axiomatic systems of Boolean logic

- It is important to understand logic laws and take them into consideration when designing logical operators for QE models
  - Few efforts have been devoted into such theoretical analysis of QE models

To do that, we must understand how logical operations are defined

Axiomatic systems of Boolean logic

- Let $\mathcal{L}$ be the set of all the valid logic formulae under a logic system, and
- $\psi_1, \psi_2, \psi_3 \in \mathcal{L}$ represent logical formulae.
- $I(\cdot)$ denotes the truth value of a logical formula.

Axiomatic systems of Boolean logic

Semantics of Boolean logic is defined by:

- The interpretation $I: \mathcal{L} \rightarrow \{0,1\}$
  - The truth value of a logical formula
    - $\mathcal{L}$: the set of all the valid logic formulae
- Logical implication
  - $\psi_1 \rightarrow \psi_2$ holds if and only if $I(\psi_2) \geq I(\psi_1)$
- The Modus Ponens inference rule
  - From ψ_1 and $\psi_1 \rightarrow \psi_2$ infer ψ_2

- A set of axioms written in Hilbert-style deductive systems
  - Define other logic connectives via logic implication ($\rightarrow$)

From Boolean Logic to Fuzzy Logic

- $I(\psi_1)$ is in [0,1]
- Axioms preserve
  - All the operations in fuzzy logic will have the same results as Boolean logic, if the operations are applied to {0,1}
- Extra axioms to define logical operations for (0,1)

How is conjunction defined

- logical implication
  - $\psi_1 \to \psi_2$ holds if and only if $I(\psi_1) \le I(\psi_2)$

The following three axioms characterize ∧:

$$\psi_1 \wedge \psi_2 \to \psi_1$$

$$\psi_1 \wedge \psi_2 \to \psi_2$$

Ensure that
$I(\psi_1 \wedge \psi_2) \le I(\psi_1)$
$I(\psi_1 \wedge \psi_2) \le I(\psi_2)$

$$(\psi_3 \to \psi_1) \to ((\psi_3 \to \psi_2) \to (\psi_3 \to \psi_1 \wedge \psi_2))$$

Ensure that $I(\psi_1 \wedge \psi_2) = 1$ if $I(\psi_1) = I(\psi_2) = 1$

They also imply commutativity and associativity of ∧!

See [Chvalovsky 2012] for proofs.

Connect to embedding model

Note:

I(*Compose*(John Lennon, Let It Be)) :=
ϕ(Compose(John Lennon, ?), *Let it Be*)

$e \in S_1 \wedge S_2 \leftrightarrow e \in S_1 \wedge e \in S_2$

- $I(\psi_1 \wedge \psi_2) \leq I(\psi_1)$

I(*Compose*(John Lennon, Let It Be) ∧ *Compose*(*Paul McCartney*, Let It Be))
≤ I(*Compose*(John Lennon, Let It Be))

Embedding model

Query — Entity

ϕ(Compose(John Lennon, ?) ∧ Compose(Paul McCartney, ?), Let It Be)
≤ ϕ(Compose(John Lennon, ?), Let It Be)

Logical Laws and Model Properties

Embedding model $\phi(q,e)$ estimates the probability that entity e answers query q

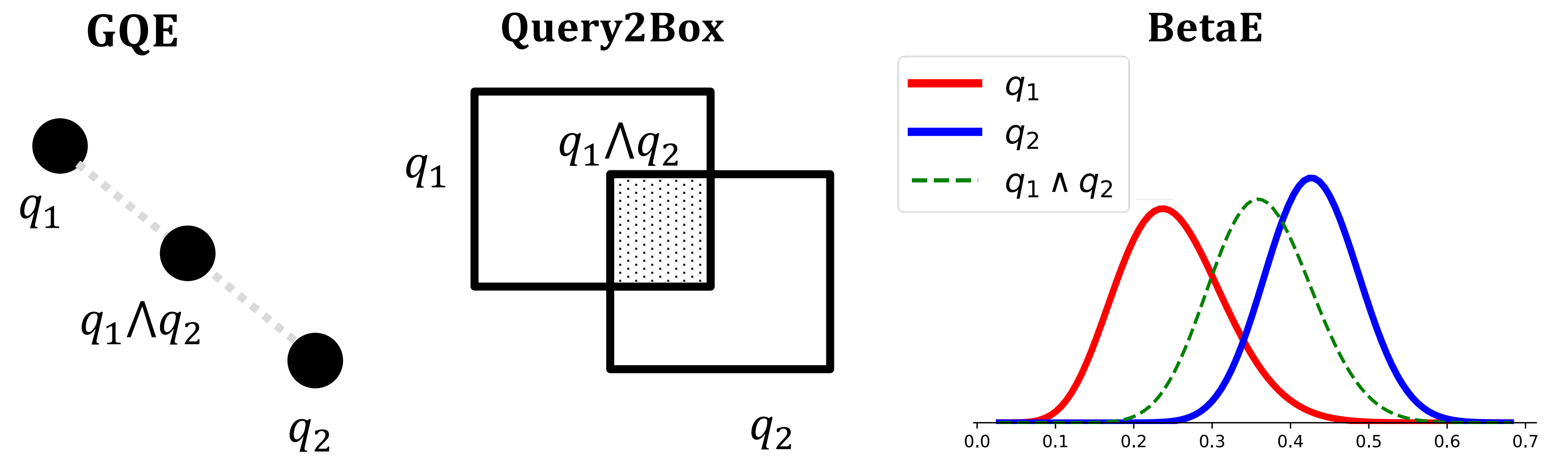
Axioms and derived logic laws in classical logic

Desired model property according to the logic law

		Logic Law	Model Property
$\wedge$	I	Conjunction Elimination $\psi_1 \wedge \psi_2 \rightarrow \psi_1$ $\psi_1 \wedge \psi_2 \rightarrow \psi_2$	$\phi(q_1 \wedge q_2, e) \leq \phi(q_1, e)$ $\phi(q_1 \wedge q_2, e) \leq \phi(q_2, e)$
	II	Commutativity $\psi_1 \wedge \psi_2 \leftrightarrow \psi_2 \wedge \psi_1$	$\phi((q_1 \wedge q_2), e) = \phi((q_2 \wedge q_1), e)$
	III	Associativity $(\psi_1 \wedge \psi_2) \wedge \psi_3 \leftrightarrow$ $\psi_1 \wedge (\psi_2 \wedge \psi_3)$	$\phi((q_1 \wedge q_2) \wedge q_3, e)$ $= \phi(q_1 \wedge (q_2 \wedge q_3), e)$

		Logic Law	Model Property
$\wedge$	I	Conjunction Elimination $\psi_1 \wedge \psi_2 \to \psi_1$ $\psi_1 \wedge \psi_2 \to \psi_2$	 $\phi(q_1 \wedge q_2, e) \leq \phi(q_1, e)$ $\phi(q_1 \wedge q_2, e) \leq \phi(q_2, e)$
	II	Commutativity $\psi_1 \wedge \psi_2 \leftrightarrow \psi_2 \wedge \psi_1$	 $\phi((q_1 \wedge q_2), e) = \phi((q_2 \wedge q_1), e)$
	III	Associativity $(\psi_1 \wedge \psi_2) \wedge \psi_3 \leftrightarrow$ $\psi_1 \wedge (\psi_2 \wedge \psi_3)$	 $\phi((q_1 \wedge q_2) \wedge q_3, e)$ $= \phi(q_1 \wedge (q_2 \wedge q_3), e)$
$\vee$	IV	Disjunction Amplification $\psi_1 \to \psi_1 \vee \psi_2$ $\psi_1 \to \psi_1 \vee \psi_1$	 $\phi(q_1, e) \leq \phi(q_1 \vee q_2, e)$ $\phi(q_2, e) \leq \phi(q_1 \vee q_2, e)$
	V	Commutativity $\psi_1 \vee \psi_2 \leftrightarrow \psi_2 \vee \psi_1$	 $\phi((q_1 \vee q_2), e) = \phi((q_2 \vee q_1), e)$
	VI	Associativity $(\psi_1 \vee \psi_2) \vee \psi_3 \leftrightarrow$ $\psi_1 \vee (\psi_2 \vee \psi_3)$	 $\phi((q_1 \vee q_2) \vee q_3, e)$ $= \phi(q_1 \vee (q_2 \vee q_3), e)$
$\neg$	VII	Involution $\neg\neg\psi_1 \to \psi_1$	 $\phi(q, e) = \phi(\neg\neg q, e)$
	VIII	Non-Contradiction $\psi_1 \wedge \neg\psi_1 \to \bar{0}$	 $\phi(q, e) \uparrow \quad \Rightarrow \quad \phi(\neg q, e) \downarrow$

Analysis of Previous Models' Capability of Preserving those Properties



	∧				∨				¬		
	Expr. (Closed)	*Com.*	*Asso.*	*Elim.*	*Expr.* (Closed)	*Com.*	*Asso.*	*Ampli.*	*Expr.* (Closed)	*Inv.*	*Non-Contra.*
GQE	✓(✓)	✓	✗	✗	✓(✗)	✓	✓	✓	✗	N/A	N/A
Query2Box	✓(✓)	✓	✓	✓	✓(✗)	✓	✓	✓	✗	N/A	N/A
BetaE	✓(✓)	✓	✗	✗	(i) DNF ✓(✗) (ii) DM ✓(✓)	✓ ✓	✓ ✓	✓ ✗	✓(✓)	✓	✗

None of previous models can satisfy all these properties

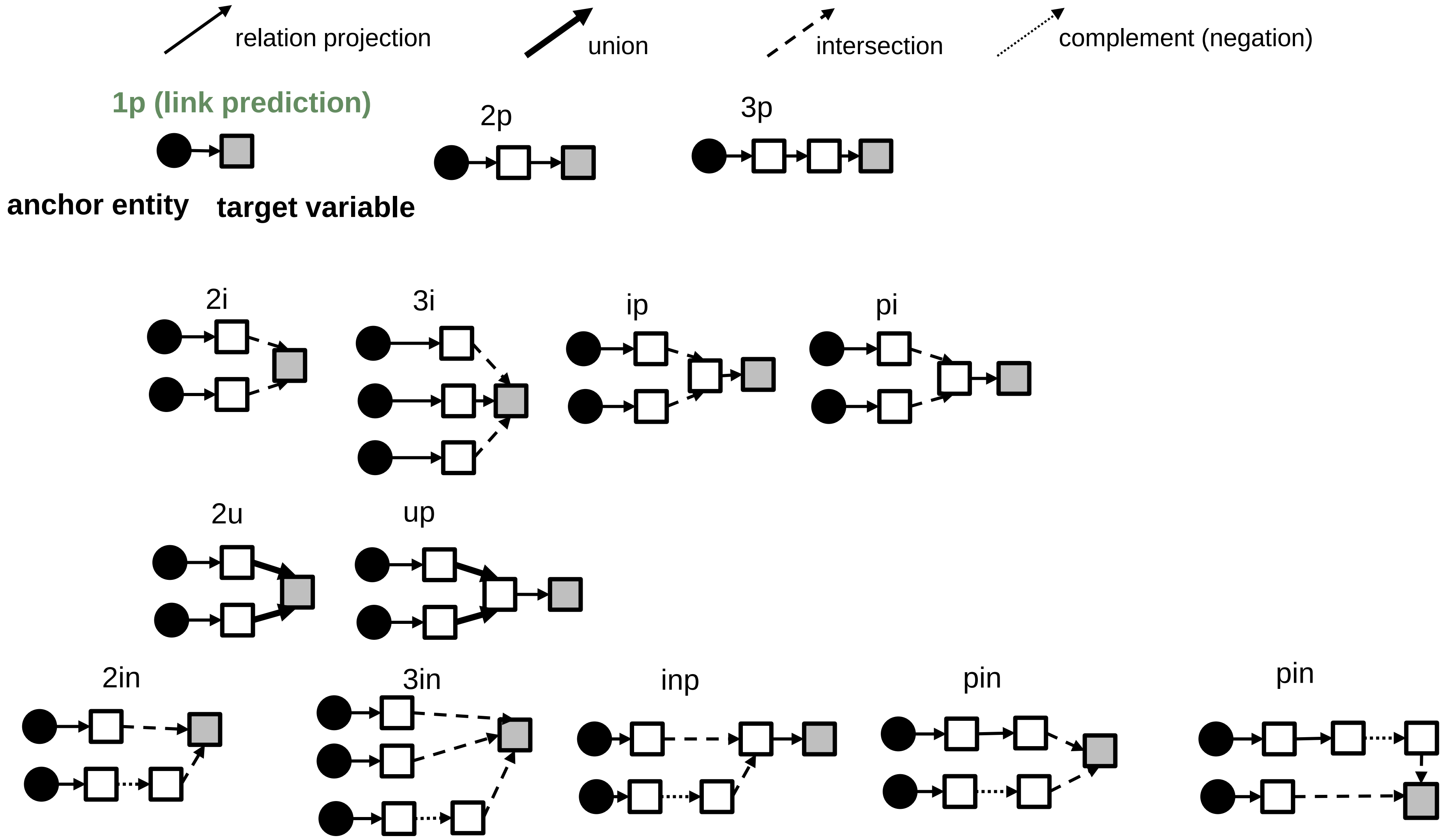
Analysis of Previous Models' Capability of Preserving those Properties

	$\wedge$				$\vee$				$\neg$		
	Expr. (Closed)	*Com.*	*Asso.*	*Elim.*	*Expr.* (Closed)	*Com.*	*Asso.*	*Ampli.*	*Expr.* (Closed)	*Inv.*	*Non-Contra.*
GQE	✓(✓)	✓	✗	✗	✓(✗)	✓	✓	✓	✗	N/A	N/A
Query2Box	✓(✓)	✓	✓	✓	✓(✗)	✓	✓	✓	✗	N/A	N/A
BetaE	✓(✓)	✓	✗	✗	(i) DNF ✓(✗) (ii) DM ✓(✓)	✓ ✓	✓ ✓	✓ ✗	✓(✓)	✓	✗
FuzzQE	✓(✓)	✓	✓	✓	✓(✓)	✓	✓	✓	✓(✓)	✓	✓

Our model can!

Dataset

FB15k-237, NELL995 with FOL queries in 14 query structures



Self-supervised training

- Logical operators do not require learning any operator specific parameters
- Significantly outperforms previous models under the same training condition (KG edges only)
  - Comparable to state-of-the-art models that are trained with extra complex query data

$$L = -\log \sigma(\phi(q,e) - \gamma) - \frac{1}{k}\sum_{i=1}^{k} \log \sigma(\gamma - \phi(q,e'))$$

Experimental Results: Trained with Atomic Queries

Table 6.8: **MRR results (%) of logical query embedding models that are trained with only link prediction.** This task tests the ability of the model to generalize to arbitrary complex logical queries, when no complex logical query data is available for training. Avg_{EPFO} and Avg_{Neg} denote the average MRR on EPFO $(\exists, \wedge, \vee)$ queries and queries containing negation respectively.

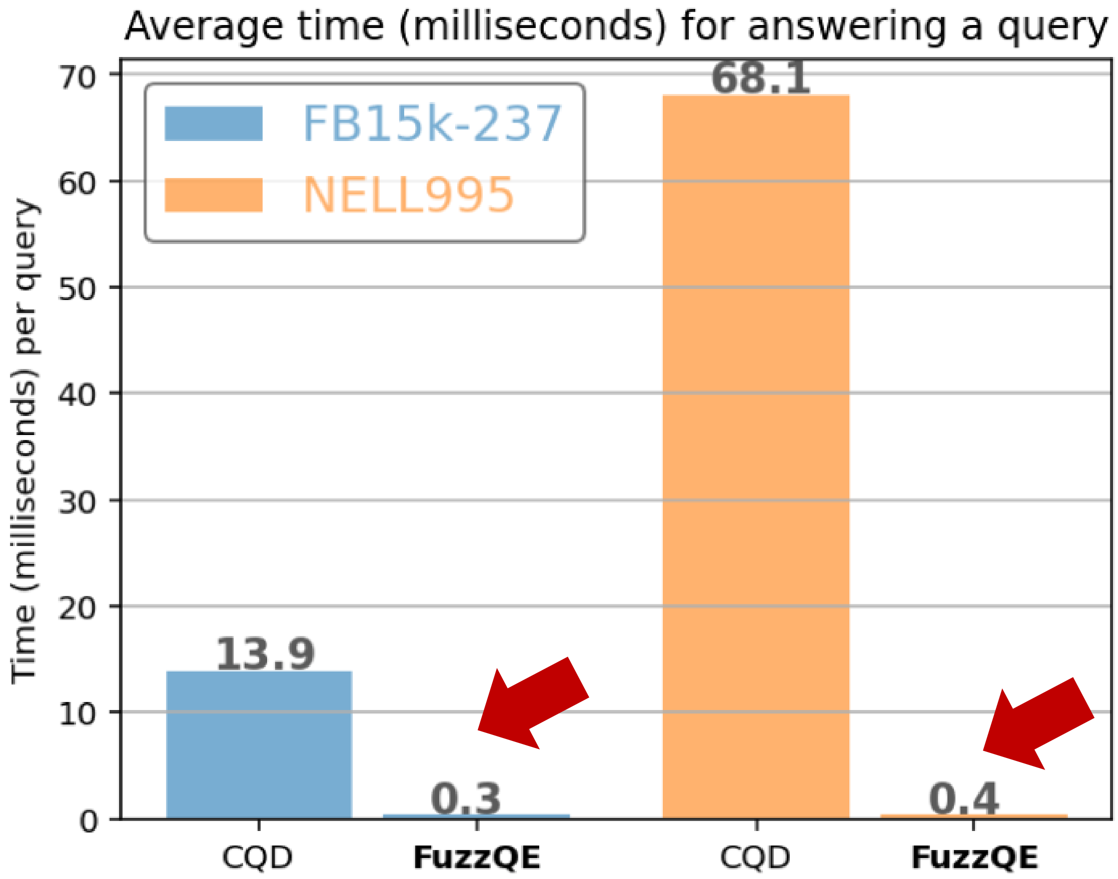
Model	Avg_{EPFO}	Avg_{Neg}	1p	2p	3p	2i	3i	pi	ip	2u	up	2in	3in	inp	pin	pni
FB15k-237																
GQE	17.7	N/A	41.6	7.9	5.4	25.0	33.6	16.3	10.9	11.9	6.2	N/A	N/A	N/A	N/A	N/A
Query2Box	18.2	N/A	42.6	6.9	4.7	27.3	36.8	17.5	11.1	11.7	5.5	N/A	N/A	N/A	N/A	N/A
BetaE	15.8	0.5	37.7	5.6	4.4	23.3	34.5	15.1	7.8	9.5	4.5	0.1	1.1	0.8	0.1	0.2
FuzzQE	**21.8**	**6.6**	**44.0**	**10.8**	**8.6**	**32.3**	**41.4**	**22.7**	**15.1**	**13.5**	**8.7**	**7.7**	**9.5**	**7.0**	**4.1**	**4.7**
NELL995																
GQE	21.7	N/A	47.2	12.7	9.3	30.6	37.0	20.6	16.1	12.6	9.6	N/A	N/A	N/A	N/A	N/A
Query2Box	21.6	N/A	47.6	12.5	8.7	30.7	36.5	20.5	16.0	12.7	9.6	N/A	N/A	N/A	N/A	N/A
BetaE	19.0	0.4	53.1	6.0	3.9	32.0	37.7	15.8	8.5	10.1	3.5	0.1	1.4	0.1	0.1	0.1
FuzzQE	**27.1**	**7.3**	**57.6**	**17.2**	**13.3**	**38.2**	**41.5**	**27.0**	**19.4**	**16.9**	**12.7**	**9.1**	**8.3**	**8.9**	**4.4**	**5.6**

Experimental Results: Trained with Additional Queries

Table 6.7: **MRR results (%) on answering FOL queries.** Report MRR results (%) on test FOL queries. Avg_{EPFO} and Avg_{Neg} denote the average MRR on EPFO queries (queries with $\exists, \wedge, \vee$ and without negation) and queries containing negation respectively. Results of GQE, Query2Box, and BetaE are taken from [81].

Type of Model	Model	Avg_{EPFO}	Avg_{Neg}	1p	2p	3p	2i	3i	pi	ip	2u	up	2in	3in	inp	pin	pni
FB15k-237																	
Query Embedding	GQE	16.3	N/A	35.0	7.2	5.3	23.3	34.6	16.5	10.7	8.2	5.7	N/A	N/A	N/A	N/A	N/A
	Query2Box	20.1	N/A	40.6	9.4	6.8	29.5	42.3	21.2	12.6	11.3	7.6	N/A	N/A	N/A	N/A	N/A
	BetaE	20.9	5.5	39.0	10.9	10.0	28.8	42.5	22.4	12.6	12.4	9.7	5.1	7.9	7.4	3.5	3.4
	FuzzQE	**24.2**	**8.5**	42.2	**13.3**	**10.2**	**33.0**	**47.3**	**26.2**	**18.9**	**15.6**	**10.8**	**9.7**	**12.6**	**7.8**	**5.8**	**6.6**
Query Optimization	CQD	21.7	N/A	**46.3**	9.9	5.9	31.7	41.3	21.8	15.8	14.2	8.6	N/A	N/A	N/A	N/A	N/A
NELL995																	
Query Embedding	GQE	18.6	N/A	32.8	11.9	9.6	27.5	35.2	18.4	14.4	8.5	8.8	N/A	N/A	N/A	N/A	N/A
	Query2Box	22.9	N/A	42.2	14.0	11.2	33.3	44.5	22.4	16.8	11.3	10.3	N/A	N/A	N/A	N/A	N/A
	BetaE	24.6	5.9	53.0	13.0	11.4	37.6	47.5	24.1	14.3	12.2	8.5	5.1	7.8	10.0	3.1	3.5
	FuzzQE	**29.3**	**8.0**	58.1	**19.3**	**15.7**	39.8	**50.3**	**28.1**	**21.8**	**17.3**	**13.7**	**8.3**	**10.2**	**11.5**	**4.6**	**5.4**
Query Optimization	CQD	28.4	N/A	**60.0**	16.5	10.4	**40.4**	49.6	28.6	20.8	16.8	12.6	N/A	N/A	N/A	N/A	N/A

Compare with CQD Regarding Inference Time



- Average time (milliseconds) for answering an FOL query on a single NVIDIA GP102 TITAN Xp (12GB) GPU.
- FB15k-237 contains 14,505 entities.
- NELL995 contains 63,361 entities, roughly 4 times the number of FB15k-237.

Outline

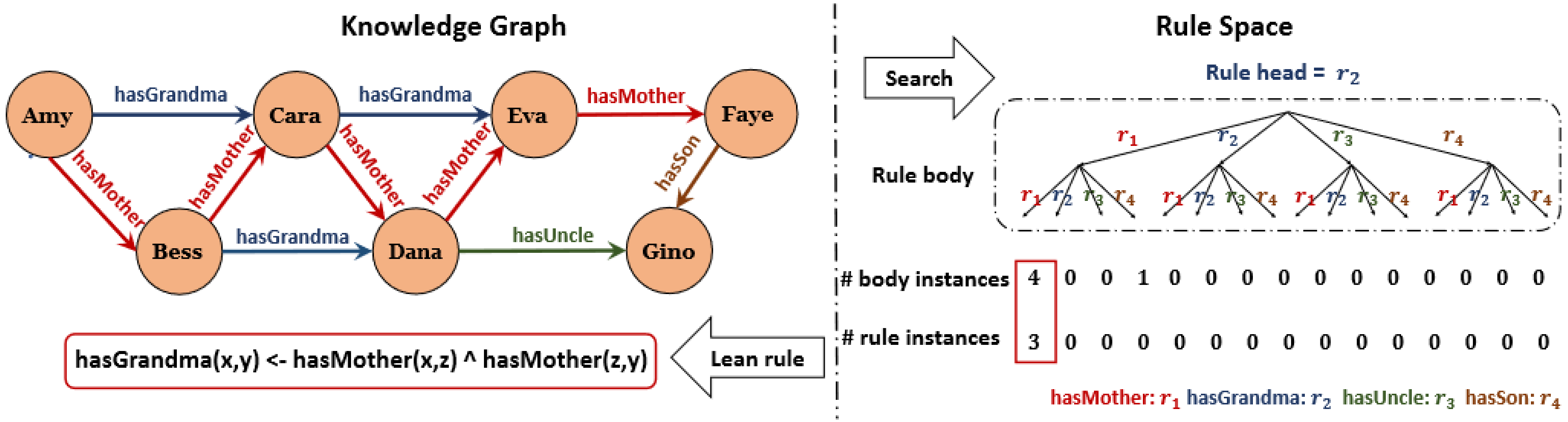
- Introduction
- Integrating Logical Rule into KGE
- KGE based Fuzzy Logic for Logical Query
- Summary

Take Away

- Logical rules provide higher-order dependency **constraints** among entities and relations
- When designing KGE-based logical query models, fuzzy logic provides a **theoretical guidance** in designing operators
- Both can reduce our demanding for data
  - Inference for cold-start entities
  - Handle query types that are never seen in the training data

Advertisement

- RLogic: Recursive Logical Rule Learning from Knowledge Graphs
  - Kewei Cheng, Jiahao Liu, Wei Wang, Yizhou Sun
  - Thursday morning



References

- [1] Guo S, Wang Q, Wang L, et al. Jointly embedding knowledge graphs and logical rules[C]//Proceedings of the 2016 conference on empirical methods in natural language processing. 2016: 192-202.
- [2] Guo S, Wang Q, Wang L, et al. Knowledge graph embedding with iterative guidance from soft rules[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).
- [3] Rocktäschel T, Singh S, Riedel S. Injecting logical background knowledge into embeddings for relation extraction[C]//Proceedings of the 2015 conference of the north American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2015: 1119-1129.
- [4] Qu M, Tang J. Probabilistic logic neural networks for reasoning[J]. arXiv preprint arXiv:1906.08495, 2019.
- [5] Zhang Y, Chen X, Yang Y, et al. Can Graph Neural Networks Help Logic Reasoning?[J]. arXiv preprint arXiv:1906.02111, 2019.
- [6] Harsha Vardhan L V, Jia G, Kok S. Probabilistic logic graph attention networks for reasoning[C]//Companion Proceedings of the Web Conference 2020. 2020: 669-673.
- [7] Kimmig A, Bach S, Broecheler M, et al. A short introduction to probabilistic soft logic[C]//Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications. 2012: 1-4.
- [8] Richardson M, Domingos P. Markov logic networks[J]. Machine learning, 2006, 62(1-2): 107-136.
- [9] Ren, H., Hu, W., and Leskovec, J. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. ICLR'2020.
- [10] Ren, H. and Leskovec, J. Beta embeddings for multi-hop logical reasoning in knowledge graphs. NeurIPS'2020.
- [11] Hamilton, W. L., Bajaj, P., Zitnik, M., Jurafsky, D., and Leskovec, J. Embedding logical queries on knowledge graphs. NeurIPS'2018.