Fuzzy Logic based Logical Query Answering on Knowledge Graphs

Xuelu Chen, Ziniu Hu, Yizhou Sun
What are knowledge graphs?

- Multi-relational graph data
- Provide structured representation for semantic relationships between real-world entities
- Serve as important knowledge sources for many real-world applications
  - Question answering systems
  - Dialogue agents
  - Recommender systems
  - ...

A triple $p(s, o)$ represents a fact, e.g.,

```
locatedIn(Eiffel Tower, Paris)
```

predicate subject object
Knowledge graph queries

• Atomic Query (Knowledge Graph Completion)
  • Given an incomplete triple, infer the missing entity
    • E.g., LocatedIn(UCLA, ?)

  Triple

  Embedding based Scoring Function

  $f(s, p, o)$

  • Related work:
    • TransE [Bordes et al. 2013], DistMult [Yang et al. 2015]; ComplEx [Trouillon et al. 2016]; RotatE [Sun et al. 2019], ...

• More Complex Logical Query
  • Given a more complicated query, infer the entity
    • E.g., “Return singers who have sung songs that were written by John Lennon or Paul McCartney but never won Grammy Award”
Knowledge graph queries

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

Logical Query

\[ q = \exists V : \forall V (\text{Compose}(\text{John Lennon}, V) \lor \text{Compose}(\text{Paul McCartney}, V)) \land \neg \text{AwardedTo}(\text{Grammy Award}, V) \land \text{SungBy}(V, V) \]

Logical Operators

- Target Variable
- Intermediate Variable

Computation Graph

(A Directed Acyclic Graph)

- **Target node**: anchor entity
- **Target node**: John Lennon
- **Target node**: Paul McCartney
- **Target node**: Grammy Award
- **Target node**: Compose
- **Target node**: Union
- **Target node**: Intersection
- **Target node**: SungBy

Graph operations:

- Compose
- AwardedTo
- SungBy
Challenging questions

• Combining Representation Learning with Logical Reasoning
  • How to represent an entity?
  • How to represent a set from a subquery?
  • 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 constraints) that logical operators should preserve?
    • Commutative, associative, etc.
  • How to train the model when additional Query-Answer pairs are not available?
Our Approach (FuzzQE)

- 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 (a set of entities)

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

Logical Query

\[ q = V \exists V ( \text{Compose}(\text{John Lennon}, V) \lor \text{Compose}(\text{Paul McCartney}, V) ) \land \neg \text{AwardedTo}(\text{ Grammy Award}, V ) \land \text{SungBy}(V, V) ]

Entity Set (Answer Set)

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

\[ q := \{ x | s(x) \text{ is true} \} \]
Bridging set and logical expressions

• Query Conjunction – Set Intersection
• Query Disjunction – Set Union
• Query Negation – Set Complement
Representing Set, Element, and Membership

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

GQE
[Hamilton et al. 2018]

Query2Box
[Ren, Hu, and Leskovec 2020]

BetaE
[Ren and Leskovec 2020]
Representing Set, Element, and Membership

• Query
  • Query as a fuzzy set, which is represented by
    • $S_q \in [0,1]^d$ (fuzzy vector)
  • Properties
    • $(1,\ldots,1) : \Omega$
    • $(0,\ldots,0) : \emptyset$
    • Subset, negation

• Entity
  • Entity as a stochastic vector
    • $P_e \in [0,1]^d$, and $\sum_i P_e(i) = 1$

• Membership
  • Embedding-based membership function
    $$\phi(q, e) = \mathbb{E}_{e \sim P_e}[e \in S_q] = \sum_{i=1}^d \Pr(e \in U_i) \Pr(U_i \subseteq S_q) = S_q^TP_e$$
Defining set operations that preserve logical laws

• Representing relation projection
  \[ S_q = P_r(p_e) = g(LN(W_r p_e + b_r)) \]
  \[ W_r = \sum_{j=1}^{K} \alpha_{rj} M_j; \quad b_r = \sum_{j=1}^{K} \alpha_{rj} v_j \]

  e.g., \textit{Compose}(John Lennon, V)

• Given the representations of two subqueries, define logical operators (set operations) with reference to product logic

  \[ q_1 \land q_2 : \quad C(S_{q_1}, S_{q_2}) = S_{q_1} \circ S_{q_2} \]
  \[ q_1 \lor q_2 : \quad D(S_{q_1}, S_{q_2}) = S_{q_1} + S_{q_2} - S_{q_1} \circ S_{q_2} \]
  \[ \neg q : \quad N(S_q) = 1 - S_q \]
More about fuzzy logic

- t-norm: conjunction (set intersection)
- t-conorm: disjunction (set union)
  - Defined by t-norm, negator: \( c(x) = 1 - x \), and De Morgan's law
Well, why are they good?

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

\[
\text{release}(\text{the Beatles, }?) \land \text{compose}(\text{John Lennon, }?) \land \text{compose}(\text{Paul McCartney, }?)
\equiv
\text{release}(\text{the Beatles, }?) \land (\text{compose}(\text{John Lennon, }?) \land \text{compose}(\text{Paul McCartney, }?))
\]

• Some previous methods (e.g. GQE) use averaging as logical operator conjunction
  • However, averaging is not associative!

\[
\frac{\frac{a+b+c}{2}}{2} \neq \frac{a+b+c}{4}
\]
Axiomatic systems of Boolean logic

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

• $\psi_1, \psi_2, \psi_3 \in \mathcal{L}$ 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 $I: \mathcal{L} \to \{0,1\}$
  - The truth value of a logical formula
    - $\mathcal{L}$: the set of all the valid logic formulae

- The Modus Ponens inference rule / Logical Implication
  - From $\psi_1$ and $\psi_1 \rightarrow \psi_2$ infer $\psi_2$
  - This defines Logical implication:
    - $\psi_1 \rightarrow \psi_2$ holds if and only if $I(\psi_2) \geq I(\psi_1)$

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

• logical implication
  • $\psi_1 \rightarrow \psi_2$ holds if and only if $I(\psi_1) \leq I(\psi_2)$

The following three axioms characterize $\land$:

- Ensure that $I(\psi_1 \land \psi_2) \leq I(\psi_1)$
- Ensure that $I(\psi_1 \land \psi_2) \leq I(\psi_2)$
- Ensure that $I(\psi_1 \land \psi_2) = 1$ if $I(\psi_1) = I(\psi_2) = 1$

They also imply commutativity and associativity of $\land$!

[Chvalovsky 2012]
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 $(0,1)$
Connect to embedding model

- \( I(\psi_1 \land \psi_2) \leq I(\psi_1) \)

\[
I(\text{Compose}(\text{John Lennon, Let It Be}) \land \text{Compose}(\text{Paul McCartney, Let It Be})) \\
\leq I(\text{Compose}(\text{John Lennon, Let It Be}))
\]

**Embedding model**

\[\phi(\text{Compose}(\text{John Lennon, } ?) \land \text{Compose}(\text{Paul McCartney, } ?), \text{Let It Be}) \leq \phi(\text{Compose}(\text{John Lennon, } ?), \text{Let It Be})\]

**Query**

**Entity**

Note:

\[
I(\text{Compose}(\text{John Lennon, Let It Be})) := \phi(\text{Compose}(\text{John Lennon, } ?), \text{Let It Be})
\]

\[
e \in S_1 \land e \in S_2 \iff e \in S_1 \land S_2
\]
**Logical Laws and Model Properties**

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

<table>
<thead>
<tr>
<th>Axioms and derived logic laws in classical logic</th>
<th>Desired model property according to the logic law</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logic Law</strong></td>
<td><strong>Model Property</strong></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
| I | Conjunction Elimination  
$\psi_1 \land \psi_2 \rightarrow \psi_1$  
$\psi_1 \land \psi_2 \rightarrow \psi_2$ | $\phi(q_1 \land q_2, e) \leq \phi(q_1, e)$  
$\phi(q_1 \land q_2, e) \leq \phi(q_2, e)$ |
| II | Commutativity  
$\psi_1 \land \psi_2 \leftrightarrow \psi_2 \land \psi_1$ | $\phi((q_1 \land q_2), e) = \phi((q_2 \land q_1), e)$ |
| III | Associativity  
$(\psi_1 \land \psi_2) \land \psi_3 \leftrightarrow \psi_1 \land (\psi_2 \land \psi_3)$ | $\phi((q_1 \land q_2) \land q_3, e) = \phi(q_1 \land (q_2 \land q_3), e)$ |
<table>
<thead>
<tr>
<th>Logic Law</th>
<th>Model Property</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conjunction</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Elimination</strong></td>
<td></td>
</tr>
<tr>
<td>I $\psi_1 \land \psi_2 \rightarrow \psi_1$ $\phi(q_1 \land q_2, e) \leq \phi(q_1, e)$</td>
<td></td>
</tr>
<tr>
<td>I $\psi_1 \land \psi_2 \rightarrow \psi_2$ $\phi(q_1 \land q_2, e) \leq \phi(q_2, e)$</td>
<td></td>
</tr>
<tr>
<td><strong>Commutativity</strong></td>
<td></td>
</tr>
<tr>
<td>II $\psi_1 \land \psi_2 \Leftrightarrow \psi_2 \land \psi_1$ $\phi((q_1 \land q_2), e) = \phi((q_2 \land q_1), e)$</td>
<td></td>
</tr>
<tr>
<td><strong>Associativity</strong></td>
<td></td>
</tr>
<tr>
<td>III $(\psi_1 \land \psi_2) \land \psi_3 \Leftrightarrow$ $\phi((q_1 \land q_2) \land q_3, e)$ $\phi(q_1 \land (q_2 \land q_3), e)$</td>
<td></td>
</tr>
<tr>
<td><strong>Disjunction</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Amplification</strong></td>
<td></td>
</tr>
<tr>
<td>IV $\psi_1 \rightarrow \psi_1 \lor \psi_2$ $\phi(q_1, e) \leq \phi(q_1 \lor q_2, e)$</td>
<td></td>
</tr>
<tr>
<td>IV $\psi_1 \rightarrow \psi_1 \lor \psi_1$ $\phi(q_2, e) \leq \phi(q_1 \lor q_2, e)$</td>
<td></td>
</tr>
<tr>
<td><strong>Commutativity</strong></td>
<td></td>
</tr>
<tr>
<td>V $\psi_1 \lor \psi_2 \leftrightarrow \psi_2 \lor \psi_1$ $\phi((q_1 \lor q_2), e) = \phi((q_2 \lor q_1), e)$</td>
<td></td>
</tr>
<tr>
<td><strong>Associativity</strong></td>
<td></td>
</tr>
<tr>
<td>VI $(\psi_1 \lor \psi_2) \lor \psi_3 \leftrightarrow$ $\phi((q_1 \lor q_2) \lor q_3, e)$ $\phi(q_1 \lor (q_2 \lor q_3), e)$</td>
<td></td>
</tr>
<tr>
<td><strong>Involution</strong></td>
<td></td>
</tr>
<tr>
<td>VII $\neg \neg \psi_1 \rightarrow \psi_1$ $\phi(q, e) = \phi(\neg \neg q, e)$</td>
<td></td>
</tr>
<tr>
<td><strong>Non-Contradiction</strong></td>
<td></td>
</tr>
<tr>
<td>VIII $\psi_1 \land \neg \psi_1 \rightarrow \bar{0}$ $\phi(q, e) \uparrow \Rightarrow \phi(\neg q, e) \downarrow$</td>
<td></td>
</tr>
</tbody>
</table>
### Analysis of Previous Models' Capability of Preserving those Properties

None of previous models can satisfy all these properties.
## Analysis of Previous Models' Capability of Preserving those Properties

<table>
<thead>
<tr>
<th></th>
<th>$\wedge$</th>
<th>$\lor$</th>
<th>$\neg$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GQE</td>
<td>✓✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Query2Box</td>
<td>✓✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BetaE</td>
<td>✓✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FuzzQE</td>
<td>✓✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Our model can!**
When complex training queries are not available

- Logical operators do not require learning any operator specific parameters

- It significantly outperforms previous query embedding models under the same training condition (using only KG edges)
  - Comparable to state-of-the-art query embedding 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

- Train with only KG Edges (No Complex 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. $\text{Avg}_{\text{EPFQ}}$ and $\text{Avg}_{\text{Neg}}$ denote the average MRR on EPFO ($\exists$, $\land$, $\lor$) queries and queries containing negation respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\text{Avg}_{\text{EPFQ}}$</th>
<th>$\text{Avg}_{\text{Neg}}$</th>
<th>1p</th>
<th>2p</th>
<th>3p</th>
<th>2i</th>
<th>3i</th>
<th>pi</th>
<th>ip</th>
<th>2u</th>
<th>up</th>
<th>2in</th>
<th>3in</th>
<th>inp</th>
<th>pin</th>
<th>pni</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB15k-237</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GQE</td>
<td>17.7</td>
<td>N/A</td>
<td>41.6</td>
<td>7.9</td>
<td>5.4</td>
<td>25.0</td>
<td>33.6</td>
<td>16.3</td>
<td>10.9</td>
<td>11.9</td>
<td>6.2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Query2Box</td>
<td>18.2</td>
<td>N/A</td>
<td>42.6</td>
<td>6.9</td>
<td>4.7</td>
<td>27.3</td>
<td>36.8</td>
<td>17.5</td>
<td>11.1</td>
<td>11.7</td>
<td>5.5</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BetaE</td>
<td>15.8</td>
<td>0.5</td>
<td>37.7</td>
<td>5.6</td>
<td>4.4</td>
<td>23.3</td>
<td>34.5</td>
<td>15.1</td>
<td>7.8</td>
<td>9.5</td>
<td>4.5</td>
<td>0.1</td>
<td>1.1</td>
<td>0.8</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td><strong>FuzzQE</strong></td>
<td><strong>21.8</strong></td>
<td><strong>6.6</strong></td>
<td><strong>44.0</strong></td>
<td><strong>10.8</strong></td>
<td><strong>8.6</strong></td>
<td><strong>32.3</strong></td>
<td><strong>41.4</strong></td>
<td><strong>22.7</strong></td>
<td><strong>15.1</strong></td>
<td><strong>13.5</strong></td>
<td><strong>8.7</strong></td>
<td><strong>7.7</strong></td>
<td><strong>9.5</strong></td>
<td><strong>7.0</strong></td>
<td><strong>4.1</strong></td>
<td><strong>4.7</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NELL995</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GQE</td>
<td>21.7</td>
<td>N/A</td>
<td>47.2</td>
<td>12.7</td>
<td>9.3</td>
<td>30.6</td>
<td>37.0</td>
<td>20.6</td>
<td>16.1</td>
<td>12.6</td>
<td>9.6</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Query2Box</td>
<td>21.6</td>
<td>N/A</td>
<td>47.6</td>
<td>12.5</td>
<td>8.7</td>
<td>30.7</td>
<td>36.5</td>
<td>20.5</td>
<td>16.0</td>
<td>12.7</td>
<td>9.6</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BetaE</td>
<td>19.0</td>
<td>0.4</td>
<td>53.1</td>
<td>6.0</td>
<td>3.9</td>
<td>32.0</td>
<td>37.7</td>
<td>15.8</td>
<td>8.5</td>
<td>10.1</td>
<td>3.5</td>
<td>0.1</td>
<td>1.4</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td><strong>FuzzQE</strong></td>
<td><strong>27.1</strong></td>
<td><strong>7.3</strong></td>
<td><strong>57.6</strong></td>
<td><strong>17.2</strong></td>
<td><strong>13.3</strong></td>
<td><strong>38.2</strong></td>
<td><strong>41.5</strong></td>
<td><strong>27.0</strong></td>
<td><strong>19.4</strong></td>
<td><strong>16.9</strong></td>
<td><strong>12.7</strong></td>
<td><strong>9.1</strong></td>
<td><strong>8.3</strong></td>
<td><strong>8.9</strong></td>
<td><strong>4.4</strong></td>
<td><strong>5.6</strong></td>
</tr>
</tbody>
</table>
## Experimental Results

- **Trained with additional complex queries**

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

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

• We propose a novel logical query embedding framework FuzzQE for answering complex logical queries on knowledge graphs.

• We present theoretical guidance for future work future research on embedding-based logical query answering models.

• Extensive experiments show the promising capability of FuzzQE on answering logical queries on KGs.
Thank you!