Contextual Graph Attention for Answering Logical Queries over Incomplete Knowledge Graphs

K-CAP 2019, Nov. 2019

Gengchen Mai¹, Krzysztof Janowicz¹, Bo Yan¹
Rui Zhu¹, Ling Cai¹, Ni Lao²

¹ STKO Lab, University of California Santa Barbara
² SayMosaic Inc., Palo Alto, CA, USA
**Introduction**

- **Knowledge Graph (KG):** a *data repository* that describes *entities* and their *relationships* across domains according to some *schema*.
- **Problem:** *Incompleteness, Sparsity*, and *Noise*

![Knowledge Graph Diagram](https://medium.com/@sderymail/challenges-of-knowledge-graph-part-1-d9ffe9e35214)

Figure From [https://medium.com/@sderymail/challenges-of-knowledge-graph-part-1-d9ffe9e35214](https://medium.com/@sderymail/challenges-of-knowledge-graph-part-1-d9ffe9e35214)
The major KG Embedding models can be classified as two categories (Wang et al. 2017):

- **Translation-based models** (e.g. TransE, TransH, and TransR)

- **Semantic matching models** (e.g. RESCAL, DisMult, and HolE)
Introduction

Training a KG embedding model over a knowledge graph (KG) $\mathcal{G} = (\mathcal{V}, \mathcal{R})$

- **Task**: link prediction and entity classification
- The **model complexity** is linear with respect to $|\mathcal{V}|$
- Dealing with more complex tasks?
**Conjunctive Graph Query (CGQ)**

Using KG Embeddings for **Conjunctive Graph Query (CGQ)**

A query \( q \in Q(G) \) that can be written as follows:

\[
q = V? \cdot \exists V_1, V_2, \ldots, V_m : b_1 \land b_2 \land \ldots \land b_n
\]

where \( b_i = r(e_k, V_l), V_l \in \{V?, V_1, V_2, \ldots, V_m\}, e_k \in \mathcal{V}, r \in \mathcal{R} \)

or \( b_i = r(V_k, V_l), V_k, V_l \in \{V?, V_1, V_2, \ldots, V_m\}, k \neq l, r \in \mathcal{R} \)

Requirements:

- Require **one variable** as the answer denotation: **Target Node**
- **No variable in the predicate position**
- Only consider the **conjunction** of graph patterns
- The dependence graph of \( q \) must be a **directed acyclic graph (DAG)**
**Conjunctive Graph Query (CGQ)**

\[
\text{?Disease}. \exists \text{?Person} : \text{AlmaMater}^{-1}(\text{UCLA}, \text{?Person}) \land \\
\text{Guest}(\text{EscapeClause}, \text{?Person}) \land \\
\text{DeathCause}(\text{?Person}, \text{?Disease})
\]

**SPARQL?**
Conjunctive Graph Query (CGQ)

\[ ?\text{Disease}. \exists ?\text{Person} : \text{AlmaMater}^{-1}(\text{UCLA, } ?\text{Person}) \land \\
\text{Guest}(\text{EscapeClause, } ?\text{Person}) \land \\
\text{DeathCause}(?\text{Person, } ?\text{Disease}) \]

KG Embedding
**Conjunctive Graph Query (CGQ)**

Using **KG Embedding** to predict the answer to a **CGQ**:

- **Projection Operation**: **Translate** from the corresponding entity nodes via different relation embeddings through different paths (triple $T_1$ and $T_2$).

- **Intersection Operation**: **Integrate** predicted embeddings for the same variable (?Person) from different paths (triple $T_1$ and $T_2$).

- Recursively use these two operators until we get the embedding for the target variable $q$.

- **Nearest neighbor search** for answer entities with $q$ by cosine similarity.
**Related Work**

- **Wang et al. (2018):** Pretrain KG embeddings and use it for CGQ
  - **lacks flexibility:** deterministic weighting approach for path embedding integration
  - **No end-to-end:** does not directly optimized on the QA objective

- **Hamilton et al. (2018):** An end-to-end model for logic query answering with an *elementwise-mean intersection operator* which treats query path **equally**
  - Fail to consider **unequal contribution from different paths**
  - Do not consider the **original KG structure**

- **Attention?**
**Attention Mechanism**

Consider **unequal contribution from different triple paths:**

- Problem for **Attention mechanism:** The **center node embedding/query embedding** is a **prerequisite** for attention score computing which is **unknown** in this case.

![Diagram illustrating unequal contribution from different triple paths.](image)
Entity Embedding

Entity embedding lookup:

\[ e_i = \frac{Z_\gamma x_i}{\| Z_\gamma x_i \|_{L2}} \quad (1) \]

- \( Z_\gamma \in \mathbb{R}^{d \times m_\gamma} \) is the type-specific embedding matrices for all entities with type \( \gamma = \Gamma(e_i) \).
**Projection Operation**

\[ e'_i = \mathcal{P}(e_i, r) = R_r e_i \]  

- \( R_r \in \mathbb{R}^{d \times d} \) is a trainable and relation-specific matrix for relation type \( r \).

\[ e'_i = \mathcal{P}(e_i, r) = R_r e_i \] (2)
**Intersection Operation**

\[ e'' = I_{CGA}(\{e'_1, e'_2, ..., e'_i, ..., e'_n\}) \] (3)

[Diagram showing the Intersection Operator and its application with examples of e1: AlmaMaster^{-1}(UCLA, ?Person) and e2: Guest(Escape Clause, ?Person).]
**Intersection Operation**

- Multi-head attention inspired by Transformer (Vaswani et al. 2017).
- **An initial intersection embedding layer** (red) is used so that center variable embedding is no longer a prerequisite.
**Model Training**

- **Original KG Training Phase:**

\[ \mathcal{L}_{KG} = \sum_{e_i \in \mathcal{V}} \sum_{e_i^\neg \in \text{Neg}(e_i)} \max(0, \Delta - \Phi(H_{KG}(e_i), e_i) + \Phi(H_{KG}(e_i), e_i^\neg)) \]  

(4)

- **Φ**: cosine similarity function.
- **H_{KG}(e_i)** indicates a new embedding \(e_i''\) for entity \(e_i\) given its 1-degree neighborhood \(N(e_i)\).
- **e_i^\neg \in \text{Neg}(e_i)** is a negative sample.

- **Logical Query-Answer Pair Training Phase:**

\[ \mathcal{L}_{QA} = \sum_{(q_i, a_i) \in S} \sum_{a_i^\neg \in \text{Neg}(q_i, a_i)} \max(0, \Delta - \Phi(q_i, a_i) + \Phi(q_i, a_i^\neg)) \]  

(5)

- **q_i**: query embedding.
- **a_i, a_i^\neg**: the embedding for the correct answer entity & negative answers.

- **The whole loss function:**

\[ \mathcal{L} = \mathcal{L}_{KG} + \mathcal{L}_{QA} \]  

(6)
Datasets

- The original **Bio** dataset (Hamilton et al. 2018)
- We constructed two datasets from publicly available **DBpedia** and **Wikidata**: **DB18**, **WikiGeo19**
- **Two metrics**: ROC AUC score and average percentile rank (APR)

Table 1: Statistics for Bio, **DB18** and **WikiGeo19** (Section 4.1). “NUM/QT” indicates the number of QA pairs per query type.

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Validation</th>
<th>Testing</th>
<th>Training</th>
<th>Validation</th>
<th>Testing</th>
<th>Training</th>
<th>Validation</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Triples</td>
<td>3,258,473</td>
<td>20,114</td>
<td>181,028</td>
<td>122,243</td>
<td>1,358</td>
<td>12,224</td>
<td>170,409</td>
<td>1,893</td>
<td>17,041</td>
</tr>
<tr>
<td># of Entities</td>
<td>162,622</td>
<td>-</td>
<td>-</td>
<td>21,953</td>
<td>-</td>
<td>-</td>
<td>18,782</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># of Relations</td>
<td>46</td>
<td>-</td>
<td>-</td>
<td>175</td>
<td>-</td>
<td>-</td>
<td>192</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># of Sampled 2-edge QA Pairs</td>
<td>1M</td>
<td>1k/QT</td>
<td>10k/QT</td>
<td>1M</td>
<td>1k/QT</td>
<td>10k/QT</td>
<td>1M</td>
<td>1k/QT</td>
<td>10k/QT</td>
</tr>
<tr>
<td># of Sampled 3-edge QA Pairs</td>
<td>1M</td>
<td>1k/QT</td>
<td>10k/QT</td>
<td>1M</td>
<td>1k/QT</td>
<td>10k/QT</td>
<td>1M</td>
<td>1k/QT</td>
<td>10k/QT</td>
</tr>
</tbody>
</table>
**Evaluation Results**

- Adding the **original KG training phase** in the model training process improves the model performance.
- Adding the **attention mechanism** further improves the model performance.
- CGA has **less learnable parameters** with better performance.
- CGA shows strong advantages over baseline models especially on query types with **hard negative sampling**.

**Table 2: Macro-average AUC and APR over test queries with different DAG structures are used to evaluate the performance.**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Bio AUC</th>
<th>APR</th>
<th>DB18 AUC</th>
<th>APR</th>
<th>WikiGeo19 AUC</th>
<th>APR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>H-Neg</td>
<td>All</td>
<td>H-Neg</td>
<td>All</td>
<td>H-Neg</td>
</tr>
<tr>
<td>Billinear[mean_simple]</td>
<td>81.65</td>
<td>67.26</td>
<td>82.39</td>
<td>70.07</td>
<td>82.85</td>
<td>64.44</td>
</tr>
<tr>
<td>Billinear[min_simple]</td>
<td>82.52</td>
<td>69.06</td>
<td>83.65</td>
<td>72.7</td>
<td>82.96</td>
<td>64.66</td>
</tr>
<tr>
<td>TransE[mean]</td>
<td>80.64</td>
<td>73.75</td>
<td>81.37</td>
<td>76.09</td>
<td>82.76</td>
<td>65.74</td>
</tr>
<tr>
<td>TransE[min]</td>
<td>80.26</td>
<td>72.71</td>
<td>80.97</td>
<td>75.03</td>
<td>81.77</td>
<td>63.95</td>
</tr>
<tr>
<td>GQE[mean]</td>
<td>83.4</td>
<td>71.76</td>
<td>83.82</td>
<td>73.41</td>
<td>83.38</td>
<td>65.82</td>
</tr>
<tr>
<td>GQE[min]</td>
<td>83.12</td>
<td>70.88</td>
<td>83.59</td>
<td>73.38</td>
<td>83.47</td>
<td>66.25</td>
</tr>
<tr>
<td>GQE+KG[mean]</td>
<td>83.69</td>
<td>72.23</td>
<td>84.07</td>
<td>74.3</td>
<td>84.23</td>
<td>68.06</td>
</tr>
<tr>
<td>GQE+KG[min]</td>
<td>83.69</td>
<td>72.23</td>
<td>84.07</td>
<td>74.3</td>
<td>84.23</td>
<td>68.06</td>
</tr>
<tr>
<td>CGA+KG+1[min]</td>
<td>84.57</td>
<td>74.87</td>
<td>85.18</td>
<td>77.11</td>
<td>84.31</td>
<td>67.72</td>
</tr>
<tr>
<td>CGA+KG+4[min]</td>
<td>85.13</td>
<td>76.12</td>
<td>85.46</td>
<td>77.8</td>
<td>84.46</td>
<td>67.88</td>
</tr>
<tr>
<td>CGA+KG+8[min]</td>
<td>85.04</td>
<td>76.05</td>
<td>85.5</td>
<td>77.76</td>
<td>84.67</td>
<td>68.56</td>
</tr>
<tr>
<td>Relative Δ over GQE</td>
<td>2.31</td>
<td>7.29</td>
<td>2.28</td>
<td>5.97</td>
<td>1.44</td>
<td>3.49</td>
</tr>
</tbody>
</table>

All and H-Neg. denote macro-averaged across all query types and query types with hard negative sampling (see Section 3.2.3).
**Evaluation Results**

- CGA **outperforms** the baseline models in almost all query types.

![APR for WikiGeo19](chart.png)

**APR for WikiGeo19**
Conclusion

- We propose an **end-to-end attention-based** logical query answering model, **contextual graph attention model (CGA)**.
- The **multi-head attention mechanism** is utilized in the intersection operator to automatically learn **different weights for different query paths**.
- Our models **outperform** the baseline models on three dataset (Bio, DB18, and WikiGeo19) despite using **less parameters**.
Future Work

- Explore ways to use our model in an **inductive learning setup**
- Consider **disjunction, negation, and filters** in query answering
- Consider **variables in the predicate position**