PAGE: Answering Pattern Queries via Knowledge Graph Embedding

S. Hong°, N. Park*, T. Chakraborty^, H. Kang´, S. Kwon·

University of Maryland College Park, USA°
University of North Carolina at Charlotte, USA*
Indraprastha Institute of Information Technology Delhi (IIIT-D), India^
Electronics and Telecommunications Research Institute, South Korea·
Graph Query Answering

A Knowledge Graph $\mathcal{G}$

A Graph Query $Q$

A child $c$ of the president $p$ of the U.S, who had visited Canada before and is a friend of a pop star $s$
Graph Query Answering [cont]

A Knowledge Graph $\mathcal{G}$

A Graph Query $Q$
Graph Query Answering [cont]

A Knowledge Graph $G$

A Graph Query $Q$

a child $C_2$ of the president $Trump$ of the U.S, who had visited Canada before and is a friend of a pop star $P_2$

Compute answers via subgraph isomorphism (matching)
Graph Query Answering: Problem

A Knowledge Graph $G$

[What if Canada is missing in $G$]

A Graph Query $Q$
Graph Query Answering: Problem [cont]

A Knowledge Graph $\mathcal{G}$

A Graph Query $Q$

[What if Canada is missing in $\mathcal{G}$]

a child $C_1$ or $C_2$ of the president Trump of the U.S, who had visited Canada before and is a friend of a pop star $P_1$ or $P_2$?

Hard to find exact answers
Knowledge Graph Embedding

A Knowledge Graph $\mathcal{G}$

An Embedding (features) $\mathbf{M}$

$\begin{pmatrix}
0.11 & 0.5 & \cdots & 0.98 \\
\vdots & \ddots & \ddots & \vdots \\
0.42 & 0.7 & \cdots & -0.3
\end{pmatrix}$
Knowledge Graph Embedding [cont]

The embedding learns vector representations of $v, r, u$ such that $v, r, u$ minimizes the energy between $v + r$ and $u$.

A Toy Knowledge Graph $G$  

An Embedding in 3-D Space

Knowledge Graph Embedding [cont]

The embedding enables to infer the unknown vertex $u$, connected to $v$ via $r$, by selecting the closest vertex around the vector $v + r$.

A Toy Knowledge Graph $\mathcal{G}$

An Embedding in 3-D Space
Knowledge Graph Embedding: Problem

A Toy Knowledge Graph $G$
[Consider single edge or unidirectional paths]

A Graph Query $Q$
[Composed of bidirectional paths]
PAGE: Graph Pattern Query Answering via Knowledge Graph Embedding

• Contributions:
  • We propose a novel method that answers a graph pattern query via knowledge graph embedding
  • Our method finds latent answers from a knowledge graph that carries incorrect or incomplete information
  • Our method learns vector representations, considering the error of bi-directional paths in a knowledge graph
PAGE: Energy of a Bi-directional Paths Query

A graph query $Q$ can be seen as a series of bi-directional path queries

$p_1$: (US) $\leftarrow$ [:IsPresidentOf]-(?p) - [:HasChild] $\rightarrow$ (?c)
$p_2$: (US) $\leftarrow$ [:IsPopstarOf]-(?s) - [:Friend] $\rightarrow$ (?c)
$p_3$: (?c) $\leftarrow$ [:Visited] - (CANADA)

A Graph Query $Q$  Bi-directional Path Queries
PAGE: Energy of a Bi-directional Paths Query [cont]

• Energy of a bi-directional path:

  • Inverse operation:
    Given a query $?x \rightarrow u$, the inverse operation is to find $x$ such that $energy(x, r, u) = 0$
    (ex. [in TransE] $x = u - r$)

  • Energy of a bidirectional path:
    Given a $h$-hop bi-directional path $p$, whose left and right ends are $u$ and $v$ with a series of intermediate relations $r_1 \ldots r_{h-1}$,
    
    $energy(p) = \begin{cases} 
    ||x + r_h - v||, & \text{if the last edge is } r_h \rightarrow v \\
    ||v + r_h - x||, & \text{if the last edge is } r_h \leftarrow v 
    \end{cases}$

    where $x$ is a vector calculated from $u$ up to $r_{h-1}$ [in TransE].
• Energy of a graph query:

  • Let $Q$ be a graph pattern query and $q$ be an candidate answer to $Q$, the energy of the graph query is defined as:

  

  $\text{energy}(q) = \sum_{p \in \text{path}(q)} \text{energy}(p)$
PAGE: Energy of a Bi-directional Paths Query [cont]

• Energy of a graph query [cont]:

• ex.

\[ p_1: \text{(US)} \leftarrow [:\text{IsPresidentOf}]-(?p) - [:\text{HasChild}] \rightarrow (?c) \]
\[ p_2: \text{(US)} \leftarrow [:\text{IsPopstarOf}]-(?s) - [:\text{Friend}] \rightarrow (?c) \]
\[ p_3: (?c) \leftarrow [:\text{Visited}] - (\text{CANADA}) \]

\[ \text{energy}(q) = e(p_1) + e(p_2) + e(p_3) \]

A Graph Query Q
PAGE: Improve Training of KGE Methods

The training datasets used in most knowledge graph embedding methods only consist of single edge (factoid) queries but PAGE need to learn the latent representation for graph queries.

A Knowledge Graph $\mathcal{G}$  
A Training Dataset for PAGE
• Sampling spanning trees:

1. Randomly choose a terminal vertex from a knowledge graph $G$
2. Perform the Join 3(b) of the FFSM [2] $e$ times so that a spanning tree that has $e$ edges can be sampled
3. Repeat the step 1 and 2 until all vertices and edges in $G$ are covered by at least $c$ different sampled spanning trees

PAGE: Improve Training of KGE Methods [cont]

- Decompose spanning trees into bi-directional paths:

A Sampled Spanning Tree

A Set of Bi-directional Paths
PAGE: Improve Training of KGE Methods [cont]

- Margin-based (Hinge) loss function

\[ \mathcal{L} = \sum_{p^+ \in \text{path}(p)} \sum_{p^-} \max(0, \gamma + e(p^+) - e(p^-)) \]

\[ \rightarrow \] Maximize the error between the true and false bi-directional paths
(ex., true path \( p^+ = (x, r_1 \ldots r_{h-1}, u) \))
false path \( p^- = (x, r_1 \ldots r_{h-1}, v) \) or \( (w, r_1 \ldots r_{h-1}, u) \)

- The optimization\(^1\) [training]

\[ \arg\min_M \mathcal{L}(M) + \lambda \mathcal{R}(M) \]

---
1. Note: we use the stochastic gradient descent (SGD) method. The details are in our paper.
Evaluation

• Experimental Setups

  • **Tasks:** 1) Factoid query answering
               2) Graph query answering

  • **Databases:** *FB15K* and *Nell186*
      • Datasets: sampled spanning trees from these databases
                 (training, testing, validating)

  • **Baseline methods:** *TransE* and *SE*\(^1\)

  • **Metrics:** Mean rank and Hits@10/100/1000

---

1. Note: as the inverse operation of the SME method cannot be defined, we use TransE and SE.
Evaluation [cont]

- Factoid Query Answering

<table>
<thead>
<tr>
<th>Database</th>
<th>Metric</th>
<th>Type</th>
<th>TransE</th>
<th>PAGE-TransE</th>
<th>SE</th>
<th>PAGE-SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB15K</td>
<td>Mean Rank</td>
<td>Micro</td>
<td>181.76</td>
<td>178.98</td>
<td>408.69</td>
<td>375.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro</td>
<td>109.02</td>
<td>106.09</td>
<td>412.04</td>
<td>364.24</td>
</tr>
<tr>
<td></td>
<td>Hits@10/100</td>
<td>Micro</td>
<td>43.4%  / 76.6%</td>
<td>44.2% / 76.2%</td>
<td>21.9% / 59.2%</td>
<td>22.3% / 59.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro</td>
<td>49.2%  / 81.3%</td>
<td>49.4% / 80.9%</td>
<td>27.6% / 62.8%</td>
<td>28.9% / 62.8%</td>
</tr>
<tr>
<td>Nell186</td>
<td>Mean Rank</td>
<td>Micro</td>
<td>885.54</td>
<td>784.02</td>
<td>3412.0</td>
<td>3752.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro</td>
<td>885.54</td>
<td>784.02</td>
<td>4492.2</td>
<td>4736.8</td>
</tr>
<tr>
<td></td>
<td>Hits@10/100</td>
<td>Micro</td>
<td>41.5%  / 74.6%</td>
<td>38.6% / 72.4%</td>
<td>10.1% / 15.75%</td>
<td>9.2% / 14.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro</td>
<td>41.5%  / 74.6%</td>
<td>38.6% / 72.4%</td>
<td>3.3% / 8.0%</td>
<td>3.0% / 7.1%</td>
</tr>
</tbody>
</table>

*at most 13% better*

*similar*
Evaluation [cont]

- Graph Query Answering

- **Graph query generation from the databases:**
  1. Merge training and testing datasets into a knowledge graph
  2. Randomly choose a vertex $v$ from the testing set
  3. Create $z$ paths by iterating the following steps $z$ times
     a) Choose a path length in between 2 and 4
     b) Randomly select a path of the chosen length starting from $v$, whose path should have at least one edge in the testing set.
  4. Convert $v$ and all intermediate vertices of the paths into variables and create a graph query $q$ from them
  5. The correct answer to the query $q$ is $v$; we are interested in finding a vertex mapped to the variable converted from $v$
Evaluation [cont]

- Graph Query Answering

<table>
<thead>
<tr>
<th>Database</th>
<th>Metric</th>
<th>Type</th>
<th>TransE</th>
<th>PAGE-TransE</th>
<th>SE</th>
<th>PAGE-SE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FB15K</strong></td>
<td>Mean Rank</td>
<td>Micro</td>
<td>1150.5</td>
<td><strong>1088</strong></td>
<td>7493.5</td>
<td>7514.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro</td>
<td>2509.9</td>
<td><strong>2362.8</strong></td>
<td>7571.4</td>
<td>7933.7</td>
</tr>
<tr>
<td></td>
<td>Hits@100/1000</td>
<td>Micro</td>
<td>18.3% / 56.7%</td>
<td><strong>25% / 60%</strong></td>
<td>1.7% / 5.0%</td>
<td>1.7% / 10%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro</td>
<td>19% / 58.7%</td>
<td><strong>24.3% / 61.7%</strong></td>
<td>2.0% / 6.0%</td>
<td>2.0% / 7.7%</td>
</tr>
<tr>
<td><strong>Nell186</strong></td>
<td>Mean Rank</td>
<td>Micro</td>
<td>38.5</td>
<td><strong>38</strong></td>
<td>4803.6</td>
<td>4960.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro</td>
<td>769.4</td>
<td><strong>491.1</strong></td>
<td>5240.3</td>
<td>5431.8</td>
</tr>
<tr>
<td></td>
<td>Hits@100/1000</td>
<td>Micro</td>
<td>64.8% / 89.4%</td>
<td><strong>66.5% / 94.6%</strong></td>
<td>15.75% / 31.4%</td>
<td>14.5% / 28.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro</td>
<td>60.2% / 80.7%</td>
<td><strong>65.4% / 87.2%</strong></td>
<td>7.9% / 23.4%</td>
<td>7.1% / 21.3%</td>
</tr>
</tbody>
</table>

*slightly improved*

*9% to 28% improvements*
Conclusions

• We propose PAGE, a novel method that answers a graph pattern query via knowledge graph embedding
• PAGE is able to find latent answers from a knowledge graph that carries incorrect or incomplete information
• PAGE improves the performances in both the factoid query (at most 13%) and graph query answering (9 to 28%) tasks
Thanks!

Sanghyun Hong
shhong@cs.umd.edu