Learning to Exploit Long-term Relational Dependencies in Knowledge Graphs

Lingbing Guo, Zequn Sun, Wei Hu*

Nanjing University, China

* Corresponding author: whu@nju.edu.cn
Knowledge graphs (KGs) store a wealth of structured facts about the real world

- A fact \((s, r, o)\): subject entity, relation, object entity

KGs are far from complete and two important tasks are proposed
Knowledge graphs

Knowledge graphs (KGs) store a wealth of structured facts about the real world

- A fact \((s, r, o)\): subject entity, relation, object entity

KGs are far from complete and two important tasks are proposed

1. **Entity alignment**: find entities in different KGs denoting the same real-world object

2. **KG completion**: complete missing facts in a single KG

- E.g., predict \(?\) in \((Tim\Berners-Lee,\ employer, \?)\) or \((?,\ employer, W3C)\)

---

**Introduction** ➤ Our method ➤ Experiments and results ➤ Conclusion
Challenges

- For KG embedding, existing methods largely focus on learning from *relational triples* of entities.

- Triple-level learning has two major limitations:
  - **Low expressiveness**
    - Learn entity embeddings from a fairly local view (i.e., 1-hop neighbors)
  - **Inefficient information propagation**
    - Only use triples to deliver semantic information within/across KGs

*Introduction ➤ Our method ➤ Experiments and results ➤ Conclusion*
Learning to exploit long-term relational dependencies

- A relational path is an **entity-relation chain**, where entities and relations appear alternately

  \[
  United \text{ Kingdom} \rightarrow \text{country}^- \rightarrow \text{Tim Berners-Lee} \rightarrow \text{employer} \rightarrow \text{W3C}
  \]

- RNNs perform well on sequential data
  - **Limitations** to leverage RNNs to model relational paths
    1. A relational path have two different types: “entity” and “relation”
      - Always appear in an alternating order
    2. A relational path is constituted by triples, but these basic structure units are overlooked by RNNs
Recurrent skipping networks

- A conditional skipping mechanism allows RSNs to **shortcut** the current input entity to let it **directly** participate in predicting its object entity.
Tri-gram residual learning

- Residual learning
  - Let $F(x)$ be an original mapping, and $H(x)$ be the expected mapping.
  - Compared to directly optimizing $F(x)$ to fit $H(x)$, it is easier to optimize $F(x)$ to fit residual part $H(x)$.
    - An extreme case, $H(x) = x$
Tri-gram residual learning

Residual learning

- Let $F(x)$ be an original mapping, and $H(x)$ be the expected mapping.
- Compared to directly optimizing $F(x)$ to fit $H(x)$, it is easier to optimize $F(x)$ to fit residual part $H(x)$
  - An extreme case, $H(x) = x$

Tri-gram residual learning

- United Kingdom → country → Tim Berners-Lee → employer → W3C
- Compared to directly learning to predict W3C by employer and its mixed context, it is easier to learn the residual part between W3C and Tim Berners-Lee
  - Because they forms a triple, and we should not overlook the triple structure in the paths

<table>
<thead>
<tr>
<th>Models</th>
<th>Optimize $F([\cdot], \text{employer})$ as</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNs</td>
<td>$F([\cdot], \text{employer}) := W3C$</td>
</tr>
<tr>
<td>RRNs</td>
<td>$F([\cdot], \text{employer}) := W3C - [\cdot]$</td>
</tr>
<tr>
<td>RSNs</td>
<td>$F([\cdot], \text{employer}) := W3C - \text{Tim Berners-Lee}$</td>
</tr>
</tbody>
</table>

[\cdot] denotes context (United Kingdom, country, Tim Berners-Lee)
Architecture

- An end-to-end framework
  1. Biased random walk sampling
     - Deep paths carry more relational dependencies than triples
     - Cross-KG paths deliver alignment information between KGs
  2. Recurrent skipping network
  3. Type-based noise contrastive estimation
     - Evaluate loss in an optimized way

Introduction ➤ Our method ➤ Experiments and results ➤ Conclusion
Experiments and results

- Entity alignment results
  - Datasets: normal & dense
  - Performed **best** on all datasets
    - Especially on the normal datasets

<table>
<thead>
<tr>
<th></th>
<th>DBP-WD</th>
<th>DBP-YG</th>
<th>EN-FR</th>
<th>EN-DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTransE</td>
<td>22.3</td>
<td>24.6</td>
<td>25.1</td>
<td>31.2</td>
</tr>
<tr>
<td>IPTransE</td>
<td>23.1</td>
<td>22.7</td>
<td>25.5</td>
<td>31.3</td>
</tr>
<tr>
<td>JAPE</td>
<td>21.9</td>
<td>23.3</td>
<td>25.6</td>
<td>32.0</td>
</tr>
<tr>
<td>BootEA</td>
<td>32.3</td>
<td>31.3</td>
<td>31.3</td>
<td>44.2</td>
</tr>
<tr>
<td>GCN-Align</td>
<td>17.7</td>
<td>19.3</td>
<td>15.5</td>
<td>25.3</td>
</tr>
<tr>
<td>TransR</td>
<td>5.2</td>
<td>2.9</td>
<td>3.6</td>
<td>5.2</td>
</tr>
<tr>
<td>TransD</td>
<td>27.7</td>
<td>17.3</td>
<td>21.1</td>
<td>24.4</td>
</tr>
<tr>
<td>ConvE</td>
<td>5.7</td>
<td>11.3</td>
<td>9.4</td>
<td>0.8</td>
</tr>
<tr>
<td>RotatE</td>
<td>17.2</td>
<td>15.9</td>
<td>14.5</td>
<td>31.9</td>
</tr>
<tr>
<td>RSNs (w/o biases)</td>
<td><strong>37.2</strong></td>
<td><strong>36.5</strong></td>
<td><strong>32.4</strong></td>
<td><strong>45.7</strong></td>
</tr>
<tr>
<td>RSNs</td>
<td><strong>38.8</strong></td>
<td><strong>40.0</strong></td>
<td><strong>34.7</strong></td>
<td><strong>48.7</strong></td>
</tr>
</tbody>
</table>
Experiments and results

Entity alignment results

- Datasets: normal & dense
- Performed **best** on all datasets
  - Especially on the normal datasets

KG completion results

- Datasets: FB15K, WN18
- Obtained **comparable** performance
  - Better than all translational models

### Hits@1

<table>
<thead>
<tr>
<th>Model</th>
<th>DBP-WD</th>
<th>DBP-YG</th>
<th>EN-FR</th>
<th>EN-DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTransE</td>
<td>22.3</td>
<td>24.6</td>
<td>25.1</td>
<td>31.2</td>
</tr>
<tr>
<td>IPTransE</td>
<td>23.1</td>
<td>22.7</td>
<td>25.5</td>
<td>31.3</td>
</tr>
<tr>
<td>JAPE</td>
<td>21.9</td>
<td>23.3</td>
<td>25.6</td>
<td>32.0</td>
</tr>
<tr>
<td>BootEA</td>
<td>32.3</td>
<td>31.3</td>
<td>31.3</td>
<td>44.2</td>
</tr>
<tr>
<td>GCN-Align</td>
<td>17.7</td>
<td>19.3</td>
<td>15.5</td>
<td>25.3</td>
</tr>
<tr>
<td>TransR</td>
<td>5.2</td>
<td>2.9</td>
<td>3.6</td>
<td>5.2</td>
</tr>
<tr>
<td>TransD</td>
<td>27.7</td>
<td>17.3</td>
<td>21.1</td>
<td>24.4</td>
</tr>
<tr>
<td>ConvE</td>
<td>5.7</td>
<td>11.3</td>
<td>9.4</td>
<td>0.8</td>
</tr>
<tr>
<td>RotatE</td>
<td>17.2</td>
<td>15.9</td>
<td>14.5</td>
<td>31.9</td>
</tr>
<tr>
<td>RSNs (w/o biases)</td>
<td>37.2</td>
<td>36.5</td>
<td>32.4</td>
<td>45.7</td>
</tr>
<tr>
<td>RSNs</td>
<td>38.8</td>
<td>40.0</td>
<td>34.7</td>
<td>48.7</td>
</tr>
</tbody>
</table>

### FB15K

<table>
<thead>
<tr>
<th>Model</th>
<th>Hits@1</th>
<th>Hits@10</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>30.5</td>
<td>73.7</td>
<td>0.46</td>
</tr>
<tr>
<td>TransR</td>
<td>37.7</td>
<td>76.7</td>
<td>0.52</td>
</tr>
<tr>
<td>TransD</td>
<td>31.5</td>
<td>69.1</td>
<td>0.44</td>
</tr>
<tr>
<td>ComplEx</td>
<td>59.9</td>
<td>84.0</td>
<td>0.69</td>
</tr>
<tr>
<td>ConvE</td>
<td>67.0</td>
<td>87.3</td>
<td>0.75</td>
</tr>
<tr>
<td><strong>RotatE</strong></td>
<td><strong>74.6</strong></td>
<td><strong>88.4</strong></td>
<td><strong>0.80</strong></td>
</tr>
<tr>
<td>RSNs (w/o cross-KG bias)</td>
<td>72.2</td>
<td>87.3</td>
<td>0.78</td>
</tr>
</tbody>
</table>
Further analysis

- RSNs vs. RNNs, RRNs [recurrent residual networks]
  - Achieved **better** results with only $1/30$ epochs
Further analysis

- RSNs vs. RNNs, RRNs [recurrent residual networks]
  - Achieved better results with only 1/30 epochs

- Random walk length
  - On all the datasets, increased steadily from length 5 to 15
Conclusion

- We studied **path-level** KG embedding learning
  1. **RSNs**: sequence models to learn relational paths
  2. **End-to-end framework**: biased random walk sampling + RSNs
  3. **Superior** in entity alignment and **competitive** in KG completion

Future work
- **Unified sequence model**: relational paths & textual information
Datasets & source code: https://github.com/nju-websoft/RSN

Acknowledgements:
- National Key R&D Program of China (No. 2018YFB1004300)
- National Natural Science Foundation of China (No. 61872172)
- Key R&D Program of Jiangsu Science and Technology Department (No. BE2018131)