IterefinE: Iterative KG Refinement Embeddings using Symbolic Knowledge

Siddhant Arora, Srikanta Bedathur, Maya Ramanath and Deepak Sharma
Indian Institute of Technology Delhi
Motivation

- KGs are often noisy and incomplete which decreases performance in downstream task
- Noise refers to various kind of errors in KG like different names for same entity, incorrect relationships and incompatible entity types
- Cleaning up of noise in KGs (KG Refinement) is usually performed using inference rules and reasoning over KGs
- New facts are inferred using KG embeddings
- GOAL : Combine ontology/inference rules with embeddings methods to improve KG refinement
Contributions

- Propose IterefinE, an iterative method to combine rule-based methods with embeddings-based methods
- Extensive experiments showing improvements upto 9% over baselines
PSL-KGI[1]

KG Embeddings

\[ L(G) = \sum_{(s,r,o,y) \in G} y \log f(s, r, o) + (1 - y) \log (1 - f(s, r, o)) \]

- **ComplEx**\(^2\) -
  \[ f(s, r, o) = e_s r_r \overline{e_o} \]

- **ConvE**\(^3\) -
  \[ f(s, r, o) = f(\text{vec}(f([\overline{e_s}; \overline{r}_r] \ast w))W)e_o \]

- **Implicit Type Supervision**\(^4\)
  \[ f(s, r, o) = \sigma(s_t \cdot r_h) \ast Y(s, r, o) \ast \sigma(o_t \cdot r_t) \]

  - \( s_t \) and \( o_t \) are implicit type embeddings of \( s \) and \( o \),
  - \( r_h \) and \( r_t \) are implicit embeddings of relation dom and range
  - \( Y \) is scoring function

---


\(^3\) T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. Convolutional 2d knowledge graph embeddings. In AAAI, 2018

\(^4\) P. Jain, P. Kumar, S. Chakrabarti, et al. Type-sensitive knowledge base inference without explicit type supervision. In ACL, 2018
Explicit Type Supervision (TypeE-X)

\[ f(s, r, o) = \sigma((s_t \| s_1) \cdot (r_h \| r_{\text{dom}})) \ast Y(s, r, o) \ast \sigma((o_t \| o_1) \cdot (r_t \| r_{\text{range}})) \]

- Here \(s_1\) and \(o_1\) are explicit entity type embeddings,
- \(r_{\text{dom}}\) and \(r_{\text{range}}\) are explicit embedding of domain and range of relation.
- The entity types, domain and range type of relation are transferred from PSL-KGI.
Algorithm Workflow

- PSL-KGI
  - Type Predictions
  - Classified Relation Triples
  - & Inferred New Triples

Filter (+ve labelled)

Filter using t2 and t3

Original KG

TypeE-X

TypeE-X Predictions

Predictions on Training Set
NELL already has noisy labels whereas for other datasets-

- Randomly sample 25% and corrupt them.
- Make 50% of the noise is type compatible and the rest is type non compatible
## Ontology Information

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DOM</th>
<th>RNG</th>
<th>SUB</th>
<th>RSUB</th>
<th>MUT</th>
<th>RMUT</th>
<th>INV</th>
<th>SAMEENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NELL</td>
<td>418</td>
<td>418</td>
<td>288</td>
<td>461</td>
<td>17K</td>
<td>48K</td>
<td>418</td>
<td>8K</td>
</tr>
<tr>
<td>FB15K-237</td>
<td>237</td>
<td>237</td>
<td>44K</td>
<td>0</td>
<td>147K</td>
<td>53K</td>
<td>44</td>
<td>20K</td>
</tr>
<tr>
<td>YAGO3-10</td>
<td>37</td>
<td>37</td>
<td>828</td>
<td>2</td>
<td>30</td>
<td>870</td>
<td>8</td>
<td>20K</td>
</tr>
<tr>
<td>WN18RR</td>
<td>11</td>
<td>11</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>66</td>
<td>0</td>
<td>20K</td>
</tr>
</tbody>
</table>

Table 4: Number of instances of each ontological component in datasets considered.

- NELL and YAGO come with rich ontology
- Type Labels are obtained for FB15k-237\(^5\) and for WN18RR\(^6\). All other rules are automatically mined for both datasets

---


\(^6\) Johannes Villmow. Transforming wn18 / wn18rr back to text., 2018.
Results

PSL KGI is hard to beat on NELL

Slightly worse on WN18RR because of very limited ontology

<table>
<thead>
<tr>
<th>Method</th>
<th>NELL</th>
<th>YAGO3-10</th>
<th>FB15K-237</th>
<th>WN18RR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+ve F1</td>
<td>-ve F1</td>
<td>wF1</td>
<td>+ve F1</td>
</tr>
<tr>
<td>ComplEx</td>
<td>0.82</td>
<td>0.58</td>
<td>0.73</td>
<td>0.94</td>
</tr>
<tr>
<td>ConvE</td>
<td>0.74</td>
<td>0.55</td>
<td>0.67</td>
<td>0.94</td>
</tr>
<tr>
<td>PSL-KGI</td>
<td>0.85</td>
<td>0.68</td>
<td>0.79</td>
<td>0.91</td>
</tr>
<tr>
<td>ConvE + ComplEx</td>
<td>0.82</td>
<td>0.58</td>
<td>0.73</td>
<td>0.95</td>
</tr>
<tr>
<td>α - ComplEx</td>
<td>0.85</td>
<td>0.68</td>
<td>0.79</td>
<td>0.94</td>
</tr>
<tr>
<td>α - ConvE</td>
<td>0.85</td>
<td>0.68</td>
<td>0.79</td>
<td>0.94</td>
</tr>
<tr>
<td>TypeE-ComplEx</td>
<td>0.86</td>
<td>0.68</td>
<td>0.79</td>
<td>0.95</td>
</tr>
<tr>
<td>TypeE-ConvE</td>
<td>0.86</td>
<td>0.67</td>
<td>0.79</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Additional Results

- Accuracy of TypeE-X methods do not vary very much with additional iterations for rich and good quality ontology.
- Adding type inferences from PSL-KGI boost performance over implicit type embeddings.
- Subclass, Domain and Range constraints are the most important however none of the individual ontological components alone show performance comparable to using all the component.
- Datasets with high quality ontology more stable in KG sizes with increasing iterations.
- Type compatible noise are harder to remove than type non compatible noise.
Thank You

Contact: siddhantaraora1806@gmail.com