Type-augmented Relation Prediction in Knowledge Graphs

Motivation

- Leverage prior type information to improve relation prediction performance
- Relation Prediction in Knowledge Graphs:
  - (Helen Mirren, ? , Chiswick)
- Prior Knowledge: type information of entities/relations
  - Helen Mirren: is a person/award_winner/actor/
  - person, place_of_birth

Type Information Encoding

- We encode the type information as prior probabilities by considering hierarchical structures among types
  - Type sets usually have an underlying hierarchy, such as the structure among types (actor, award_winner, person):
    - $H_1 = \text{person/actor}$
    - $H_2 = \text{person/award_winner}$
    - $H_3 = \text{person}$

- Hierarchy-based type weights
  - We define hierarchy-based type weights to assign different weights to types based on their locations in the hierarchy
  - We hypothesize that types of more specific semantic meaning are more helpful, and higher weights are automatically assigned to these types
  - Example, given three hierarchies $H_1, H_2$ and $H_3$, we have type weights:
    - $w_3(\text{person}) = \min(0.27, 0.27, 1) = 0.27$
    - $w_4(\text{actor}) = 0.73$
    - $w_2(\text{award_winner}) = 0.73$

- Type-based prior probability
  - Given a triple $(e_0, r, e_1) \in G$, we define two similarity score $s(e_0, r)$ and $s(e_1, r)$ based on the correlation between type sets
  - The prior probability $p(T(e_0, e_1, r))$ is then defined as
    - $p(T(e_0, e_1, r)) = \frac{s(e_0, r)}{\sum_{r \in R} s(e_0, r) \times s(e_1, r)}$
    - Where $\{e_0, e_1\}$ is the type information for entity pair $(e_0, e_1)$ and the relation set $R$
    - The higher the correlation between type sets, the higher the prior probability of the relation

Embedding-based Models

- Embedding-based models learn representations of relations and entities by minimizing the distance $f_p(e_0, e_1, r)$ in a continuous embedding space
- Given the learned embeddings, we compute the likelihood by taking the exponential
  - $p(e_0, e_1, r) = \exp\left(f_p(e_0, e_1, r)\right)$
  - The lower the distance, the lower the likelihood

Type Information Integration

- Type Information Integration is performed based on probabilities
  - For each pair of entities $(e_0, e_1)$, the posterior probability is
    - $p(r|e_0, e_1, T(e_0, e_1, r)) = p(e_1|r, e_0, T(e_0, e_1, r))$

Experiments

- Evaluation of the TaRP model
  - Baseline 1: embedding-based model trained on observed triples
  - Baseline 2: embedding-based model trained on observed triples + type triples

<table>
<thead>
<tr>
<th>Models</th>
<th>F1@10</th>
<th>Hits@10</th>
<th>MR</th>
<th>Hits@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>92.84</td>
<td>99.90</td>
<td>1.08</td>
<td>76.30</td>
</tr>
<tr>
<td>RotatE</td>
<td>92.8</td>
<td>99.90</td>
<td>1.08</td>
<td>76.30</td>
</tr>
<tr>
<td>QuatE</td>
<td>92.8</td>
<td>99.90</td>
<td>1.08</td>
<td>76.30</td>
</tr>
<tr>
<td>DB11K-174</td>
<td>92.8</td>
<td>99.90</td>
<td>1.08</td>
<td>76.30</td>
</tr>
</tbody>
</table>

Conclusions

- We achieve significantly better performance by leveraging type information compared to SoTAs on four benchmark datasets
- Our proposed approach is effective in integrating type information
- In the paper, we also show that our method is more data efficient. Through cross-dataset evaluation, we show that type information extracted from a specific dataset can generalize well to different datasets