Appendices

A Negative Sampling

We train CoPER by minimizing the binary cross-entropy between for predicted distribution over answer entities. However, one challenge is that the training data only contains triples of the form \((e_s, r, e_t)\) and there is no clear way to obtain negative samples of the form “this entity is not a correct answer to this question.” A common approach to address this challenge is to consider all answers that appear in the training set as correct answers and all other entities in the KG as wrong answers. However, this approach exhibits two issues. First, it can become very expensive for large KGs as all the entities are involved in the computation of the loss function for each training example. In addition to that, it is not necessarily correct because some of the entities treated as incorrect answers during training are the very answers we are asked to infer at test time. To alleviate these two issues, we use an alternative approach based on that of Bordes et al. (2013) and Yang et al. (2015). Instead of considering all possible alternative answers as wrong, we uniformly sample a fixed number of alternative entities, for each positive triple, and use them as negative training examples. Our experiments show that this helps boost performance and also significantly improve efficiency.

B Similar Work Comparison Evaluations

In this section we show additional comparisons between CoPER, ConvE, MINERV A and two methods that allow for multiplicative interactions: TransR (Lin et al. 2015) and TransD (Ji et al. 2015). Further details on the architectures of these models are described in our Related Work Section. The results for TransR and TransD are as reported at https://github.com/thunlp/OpenKE, and all numbers represent Hits@1. We note that results for CTransR (Lin et al. 2015) are unavailable due to its unmaintained or unavailable implementation. While CoPER-MINERVA performs similarly to TransR, CoPER-ConvE significantly outperforms all other models on both datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>WN18RR</th>
<th>FB15k237</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransR</td>
<td>WN18RR</td>
<td>51.9</td>
<td>51.1</td>
</tr>
<tr>
<td>TransD</td>
<td>WN18RR</td>
<td>50.8</td>
<td>48.7</td>
</tr>
<tr>
<td>CoPER-ConvE</td>
<td>WN18RR</td>
<td>56.12</td>
<td>62.97</td>
</tr>
<tr>
<td>CoPER-MINERVA</td>
<td>WN18RR</td>
<td>50.99</td>
<td>50.39</td>
</tr>
<tr>
<td>ConvE</td>
<td>WN18RR</td>
<td>52.27</td>
<td>60.83</td>
</tr>
<tr>
<td>MINERVA</td>
<td>WN18RR</td>
<td>51.3</td>
<td>56.4</td>
</tr>
</tbody>
</table>

Table 1: Overview of Hits@1 comparisons between CoPER models, TransR and TransD on FB15k237 and WN18RR.

1Its original implementation: https://github.com/thunlp/KB2E is no longer maintained and we were unable to train.
2It is missing from the official repository: https://github.com/thunlp/OpenKE.

C Relation Similarity Visualizations

Recall that one of our purported benefits of the contextual parameter generators described in our Parameter Generator Network Section was that relation information could be shared through the generator using approaches such as \(g_{\text{linear}}\), enabling relations to leverage knowledge from their similar counterparts. To illustrate this, we analyze the resultant relation similarities after training CoPER-ConvE with \(g_{\text{linear}}\) on both FB15k-237 and NELL-995. Figure 2 showcases the relation similarity clusters formed by FB15k-237’s relations. From the TSNE plot, we observe that many distinct relation clusters form through training. Moreover, among the 7 labeled clusters we randomly choose, we observe that their relations are all semantically similar to one another. Similarly, Figure 1 illustrates several observed relation similarity groupings. Specifically, the plot visualizes the pairwise similarities between each of the 200 relations in NELL, color-coded by their cosine distance from one another. Before plotting, each relation is grouped together according to manually defined “types” based on our analysis of the data. Based on the heatmap we observe several clear block diagonal structures, which indicate that relations are more similar to those of the same type than others from different types. For ease of understanding, we have labeled several blocks with their respective relation group “type”. All materials relevant to our “type” assignment and code for this visualization can be found in our repository at https://github.com/otiliastr/coper. In addition, we display the top six most and least similar NELL-995 relations according to their cosine distance in Tables 2 and 3 respectively. Comparable to our findings in FB15k-237, we observe that the most similar relations convey comparable semantic information, while the least similar describe disparate data.
### Freebase Relations

- /film/film/language
- /people/marriage/spouse
- /base/popstar/breakup/participant
- /base/popstar/dated/participant
- /location/religion_percentage/religion
- /organization/non_profit_registering_agency
- /people/person/place_of_birth
- /people/placed_lived/location
- /music/artist/origin
- /people/ethnicity/languages_spoken
- /people/person/languages
- /film/film/written_by
- /tv/tv_program/program_creator
- /film/film/produced_by

### Locations

- /location/us_county/county_seat
- /location/country/capital

### Linguistics

- /film/film/language

### Administrative Centers

- /people/marriage/location_of_ceremony
- /location/us_county/county_seat
- /location/country/capital

### Sports

- /baseball/baseball_roster_position/position
- /soccer/football_roster_position/position
- /basketball/basketball_roster_position/position

### References


