Troubleshooting

Loading a Model from an Old Version of PyKEEN

If your model was trained on a different version of PyKEEN, you might have difficulty loading the model using `torch.load('trained_model.pkl')`.

This could be due to one or both of the following:

1. The model class structure might have changed.
2. The model weight names might have changed.

Note that PyKEEN currently cannot support model migration. Please attempt the following steps to load the model.

If the model class structure has changed

You will likely see an exception like this one: `ModuleNotFoundError: No module named ...`

In this case, try to instantiate the model class directly and only load the state dict from the model file.

1. Save the model's `state_dict` using the version of PyKEEN used for training:

```
import torch
from pykeen.pipeline import pipeline

result = pipeline(dataset="Nations", model="RotatE")
torch.save(result.model.state_dict(), "v1.7.0/model.state_dict.pt")
```

2. Load the model using the version of PyKEEN you want to use. First instantiate the model, then load the state dict:

```python
import torch
from pykeen.datasets import get_dataset
from pykeen.models import RotatE

dataset = get_dataset(dataset="Nations")
model = RotatE(triples_factory=dataset.training)
state_dict = torch.load("v1.7.0/model.state_dict.pt")
model.load_state_dict(state_dict)
```

If the model weight names have changed

You will likely see an exception similar to this one:

```
RuntimeError: Error(s) in loading state_dict for RotatE:
Missing key(s) in state_dict: "entity_representations.0._embeddings.weight",
"relation_representations.0._embeddings.weight".
Unexpected key(s) in state_dict: "regularizer.weight", "regularizer.regularization_term",
"entity_embeddings._embeddings.weight", "relation_embeddings._embeddings.weight".
```

In this case, you need to inspect the state-dict dictionaries in the different version, and try to match the keys. Then modify the state dict accordingly before loading it. For example:

```python
import torch
from pykeen.datasets import get_dataset
from pykeen.models import RotatE

dataset = get_dataset(dataset="Nations")
model = RotatE(triples_factory=dataset.training)
state_dict = torch.load("v1.7.0/model.state_dict.pt")
# these are some example changes in weight names for RotatE between two different pykeen
versions
for old_name, new_name in [
    (
        "entity_embeddings._embeddings.weight",
        "entity_representations.0._embeddings.weight",
    ),
    (
        "relation_embeddings._embeddings.weight",
        "relation_representations.0._embeddings.weight",
    ),
]:
    state_dict[new_name] = state_dict.pop(old_name)
# in this example, the new model does not have a regularizer, so we need to delete
corresponding data
for name in ["regularizer.weight", "regularizer.regularization_term"]:
    state_dict.pop(name)
model.load_state_dict(state_dict)
```

Warning

Even if the state dict can be loaded, there is still a risk that the the weights are used differently. This can lead to a difference in model behavior. To be sure that the model is still functioning the same way, you should also check some model predictions and inspect *how* the model definition has changed.