

Solver>>>base.py

```
1. L48      import torch.nn as nn
2. L49-72   class LabelDifference(nn.Module):
3. L73-84   class FeatureSimilarity(nn.Module):
4. L85-120  class NERLoss(nn.Module):
5. L301     labels = torch.stack((prediction.index_ref, prediction.index_pos,
    prediction.Z_target), dim=1)
6. L302     self.criterion_NER = NERLoss()
7. L303     features = torch.stack((prediction.reference, prediction.positive), dim=1)
8. L304     loss = self.criterion_NER(features, labels)
```

Solver>>>single_session.py

```
1. L67     index_ref = batch.index_ref
2. L68     index_pos = batch.index_pos
3. L69     Z_target = batch.Z_target
4. L70     return ceбра.data.Batch(ref, pos, neg, index_ref, index_pos, Z_target)
```

Data>>>datatypes.py

```
1. L49     __slots__ = ["reference", "positive", "negative", "index_ref", "index_pos",
    "Z_target", "index", "index_reversed"]
2. L55     index_ref,
3. L56     index_pos,
4. L57     Z_target,
5. L63     self.index_ref = index_ref
6. L64     self.index_pos = index_pos
7. L65     self.Z_target = Z_target
```

Data>>>single_session.py

```
1. L69     XY_position = 1*self.continuous_index[:, 0:2]
2. L70     Z_target = 1*self.continuous_index[:, 2][index.reference]
3. L75     index_ref = XY_position[index.reference, 0],
4. L76     index_pos = XY_position[index.reference, 1],
5. L77     Z_target = Z_target
```

Distributions>>>index.py

```
1. L74     self.index = samples[:, 0:2]
2. L92     query = query[:, 0:2]
```