import matchzoo as mz

train\_pack = mz.datasets.wiki\_qa.load\_data('train', task='ranking')

valid\_pack = mz.datasets.wiki\_qa.load\_data('dev', task='ranking')

predict\_pack = mz.datasets.wiki\_qa.load\_data('test', task='ranking')

preprocessor = mz.preprocessors.DSSMPreprocessor()

train\_processed = preprocessor.fit\_transform(train\_pack)

valid\_processed = preprocessor.transform(valid\_pack)

ranking\_task = mz.tasks.Ranking(loss=mz.losses.RankCrossEntropyLoss(num\_neg=4))

ranking\_task.metrics = [

mz.metrics.NormalizedDiscountedCumulativeGain(k=3),

mz.metrics.NormalizedDiscountedCumulativeGain(k=5),

mz.metrics.MeanAveragePrecision()

]

model = mz.models.DSSM()

model.params['input\_shapes'] = preprocessor.context['input\_shapes']

model.params['task'] = ranking\_task

model.params['mlp\_num\_layers'] = 3

model.params['mlp\_num\_units'] = 300

model.params['mlp\_num\_fan\_out'] = 128

model.params['mlp\_activation\_func'] = 'relu'

model.guess\_and\_fill\_missing\_params()

model.build()

model.compile()

train\_generator = mz.PairDataGenerator(train\_processed, num\_dup=1, num\_neg=4, batch\_size=64, shuffle=True)

valid\_x, valid\_y = valid\_processed.unpack()

evaluate = mz.callbacks.EvaluateAllMetrics(model, x=valid\_x, y=valid\_y, batch\_size=len(pred\_x))

history = model.fit\_generator(train\_generator, epochs=20, callbacks=[evaluate], workers=5, use\_multiprocessing=False)