import sys

sys.path.append('..')

import fewshot\_re\_kit

import torch

from torch import autograd, optim, nn

from torch.autograd import Variable

from torch.nn import functional as F

class Pair(fewshot\_re\_kit.framework.FewShotREModel):

def \_\_init\_\_(self, sentence\_encoder, hidden\_size=230):

fewshot\_re\_kit.framework.FewShotREModel.\_\_init\_\_(self, sentence\_encoder)

self.hidden\_size = hidden\_size

# self.fc = nn.Linear(hidden\_size, hidden\_size)

self.drop = nn.Dropout()

def forward(self, batch, N, K, total\_Q):

'''

support: Inputs of the support set.

query: Inputs of the query set.

N: Num of classes

K: Num of instances for each class in the support set

Q: Num of instances in the query set

'''

logits = self.sentence\_encoder(batch)

logits = logits.view(-1, total\_Q, N, K, 2)

logits = logits.mean(3) # (-1, total\_Q, N, 2)

logits\_na, \_ = logits[:, :, :, 0].min(2, keepdim=True) # (-1, totalQ, 1)

logits = logits[:, :, :, 1] # (-1, total\_Q, N)

logits = torch.cat([logits, logits\_na], 2) # (B, total\_Q, N + 1)

\_, pred = torch.max(logits.view(-1, N + 1), 1)

return logits, pred