#coding:utf-8

import tensorflow as tf

from data\_helper\_vocab import VocabHelperReverse

from data\_helper import DataHelper

from models\_InceptionE import InceptionE

from trainer import Trainer

from data\_evaluate import evaluate\_model

from data\_utils.params import Params

data\_dir="WN18RR"

# Parameters

kb\_params = {

"share\_emb":False,

#data

"data\_dir":data\_dir,

"entity\_vocab\_size": 40943,

"relation\_vocab\_size": 11,

"triple\_num":86835,

#model

"emb\_dim":100,

#loss

"l2\_reg\_lambda":1e-5,

"gamma":5,

#train

"optimizer":tf.train.AdamOptimizer,

"keep\_prob":0.6,

"batch\_size":256,

"lr": 0.0002,

"lr\_decay": 0.95,

"lr\_decay\_step": 1,

"warm\_up\_step": 5

}

params = Params(data\_dir)

if \_\_name\_\_=="\_\_main\_\_":

params=Params(data\_dir)

params.update(kb\_params)

vocabHelper=VocabHelperReverse(params=params)

#datasets

valid\_dataHelper=DataHelper(dataPath=params.valid\_dataPath, params=params, vocabHelper=vocabHelper)

test\_dataHelper = DataHelper(dataPath=params.test\_dataPath, params=params, vocabHelper=vocabHelper)

train\_dataHelper=DataHelper(dataPath=params.train\_dataPath, params=params, vocabHelper=vocabHelper)

entities = vocabHelper.entity2id.keys()

entity\_embeddings=vocabHelper.entity\_embeddings

relation\_embeddings=vocabHelper.relation\_embeddings

with tf.Session() as sess:

#model

model=InceptionE(params,

entity\_embedding=vocabHelper.entity\_embeddings,

relation\_embedding=vocabHelper.relation\_embeddings)

#train

trainer=Trainer(model, params)

sess.run(tf.global\_variables\_initializer())

# trainer.restore\_last\_session(sess)

trainer.train(sess,data\_helper=train\_dataHelper,eval\_data\_helper=valid\_dataHelper,test\_data\_helper=test\_dataHelper,iter\_num=100)

#predict

evaluate\_model(sess,trainer,test\_dataHelper)