# ECISA2021-myeclipse

## 文件说明

|bert # bert的预训练模型存储位置

|bert\_model.ckpt

|bert\_model.config

|vocab.txt

|cache\_file # 用来保存模型输出概率值的文件，模型融合使用

|...

|data # 用来保存训练数据的文件

|SMP2019\_ECISA\_Train.xml

|SMP2019\_ECISA\_Dev.xml

|SMP2019\_ECISA\_Test.xml

|ECISA2021-Test.xml

|nezha # nezha base的预训练模型存储位置

|...

|NEZHA-Large-WWM # nezha的预训练模型存储位置

|...

|outputs # 模型输出保存到的文件夹

|...

|results # 存放模型（可提交）输出的文件夹

|...

|roberta # roberta large的预训练模型存储位置

|...

|utils # 工具包

|common\_function.py # 一些模型需要的公共方法（仅模型3使用）

|evaluate.py # 用于模型验证的一些公共方法（仅模型3使用）

|model\_predict.py # 用于模型预测的一些公共方法（仅模型3使用）

|add\_two\_files.py # 将多模型结果进行平均加权，输出最终结果

|bert\_dense\_segment\_v1\_ema.py # 模型1

|bert\_dense\_segment\_v2\_ema\_FGM.py # 模型2

|config.py # 配置文件

|data\_utils.py # 数据处理工具类

|NEZHA\_dpot\_dense\_ema\_rdrop.py # 模型3

|predict.py # 预测文件（模型1，模型2使用）

|predict\_final.py # 预测文件（模型3使用）

|requirements.txt # 环境文件

|train.py # 训练文件（模型1，模型2使用）

|train\_.py # 训练文件（模型3使用）

预训练权重下载

Bert base：

Google https://storage.googleapis.com/bert\_models/2018\_11\_03/chinese\_L-12\_H-768\_A-12.zip

roberta large:

Brightmart https://drive.google.com/open?id=1W3WgPJWGVKlU9wpUYsdZuurAIFKvrl\_Y

nezha base:

Huawei https://drive.google.com/drive/folders/1tFs-wMoXIY8zganI2hQgDBoDPqA8pSmh?usp=sharing

nezha large:

Huawei https://drive.google.com/drive/folders/1LOAUc9LXyogC2gmP\_q1ojqj41Ez01aga?usp=sharing

## 使用方法

1.初始化环境 `pip install -r requirements.txt`

2.下载预训练权重并存放到对应的文件夹下

3.开始训练，

- 模型1 模型2采用`train.py`进行训练 。

python train.py

-model\_name bert\_dense\_segment\_v2\_ema\_FGM

-save\_name test\_model.weights

-lr 1e-5

-bsz 32

-epoch 10

-dpot 0.2

-gpu 0,1

-nezha True

-roberta True

- 模型3 采用`train\_.py`进行训练。

python train\_.py

-model\_name NEZHA\_dpot\_dense\_ema\_rdrop

-save\_name test\_model.weights

-lr 1e-5

-bsz 16

-epoch 10

-gpu 0,1

-ema True

-rdrop True

-rmask 0

## 其他说明

最终的结果为6个模型的融合，分别为

- **nezha base（模型1）**

`python train.py -model\_name bert\_dense\_segment\_v2\_ema\_FGM -save\_name nezha\_ema\_FGM\_dp\_0.2\_time0.weights -gpu 0 -lr 1e-5 -epoch 10 -bsz 32 -dpot 0.2 -nezha True`

`python predict.py -model\_name bert\_dense\_segment\_v2\_ema\_FGM -save\_name nezha\_ema\_FGM\_dp\_0.2\_time0.weights -gpu 0 -nezha True`

**- bert base （模型1）**

`python train.py -model\_name bert\_dense\_segment\_v2\_ema\_FGM -save\_name bert\_ema\_FGM\_dp\_0.2\_time4.weights -gpu 0 -lr 1e-5 -epoch 10 -bsz 32 -dpot 0.2`

`python predict.py -model\_name bert\_dense\_segment\_v2\_ema\_FGM -save\_name bert\_ema\_FGM\_dp\_0.2\_time4.weights -gpu 0`

**- bert base （模型1）**

`python train.py -model\_name bert\_dense\_segment\_v2\_ema\_FGM -save\_name bert\_ema\_FGM\_dp\_0.2\_time2.weights -gpu 0 -lr 1e-5 -epoch 10 -bsz 32 -dpot 0.2`

`python train.py -model\_name bert\_dense\_segment\_v2\_ema\_FGM -save\_name bert\_ema\_FGM\_dp\_0.2\_time2.weights -gpu 0`

**- bert base （模型2）**

`python train.py -model\_name bert\_dense\_segment\_v1\_ema -save\_name segment\_v1\_ema\_time0.weights -gpu 0 -lr 1e-5 -epoch 10 -bsz 32 -dpot 0.2`

`python predict.py -model\_name bert\_dense\_segment\_v1\_ema -save\_name segment\_v1\_ema\_time0.weights -gpu 0`

**- nezha large（模型3）**

`python train\_.py -model\_name NEZHA\_dpot\_dense\_ema\_rdrop -save\_name nezha\_dpot\_dense\_ema.weghts -bsz 16 -epoch 10 -lr 1e-5 -ema True -gpu 0 -rdrop True -rmask 0`

`python predict\_final.py -model\_name NEZHA\_dpot\_dense\_ema\_rdrop -save\_name nezha\_dpot\_dense\_ema.weghts -gpu 0`

**- roberta large（模型2）**

`python train.py -model\_name bert\_dense\_segment\_v1\_ema -save\_name segment\_large\_v1\_ema\_time1.weights -gpu 0 -lr 1e-5 -epoch 10 -bsz 32 -dpot 0.2 -roberta True`

`python predict.py -model\_name bert\_dense\_segment\_v1\_ema -save\_name segment\_large\_v1\_ema\_time1.weights -gpu 0 `

获取到6个模型的predict输出后，调用`add\_two\_files.py`（需要更改对应文件名）来计算六个模型的加权平均作为最终结果。