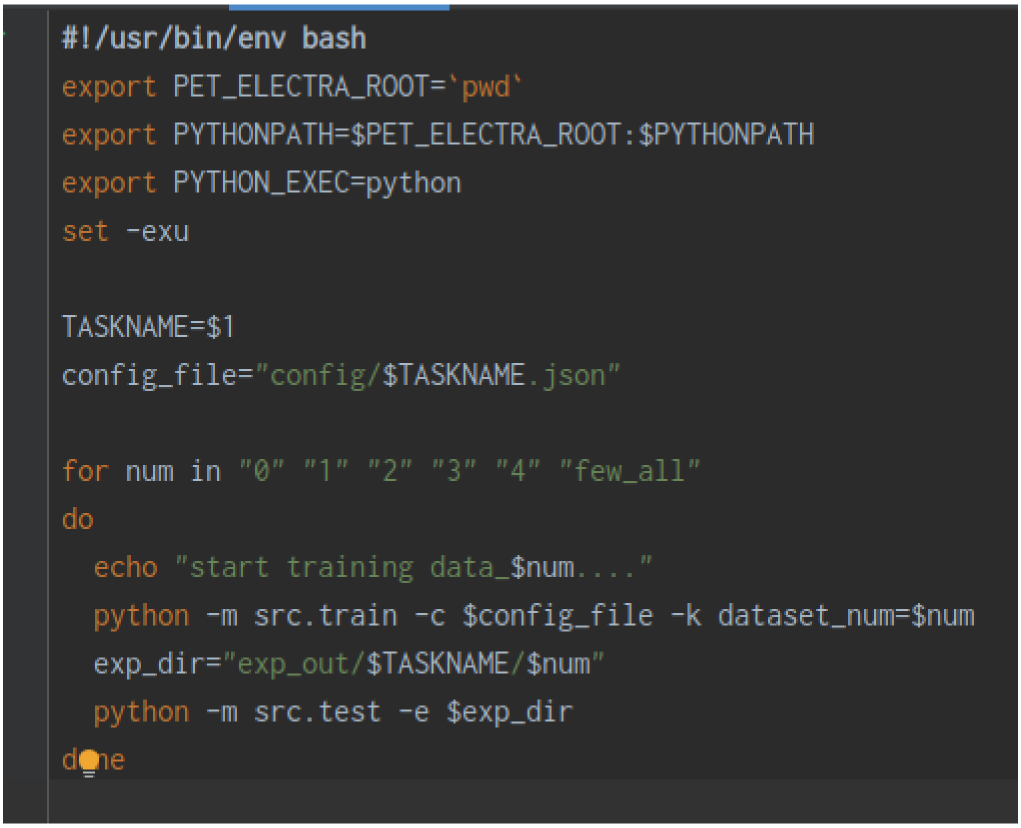
1、一键运行train test

```
bash bin/fewclue.sh {task_name}
```

2、运行逻辑

- fewclue.sh 调用train和test相应脚本



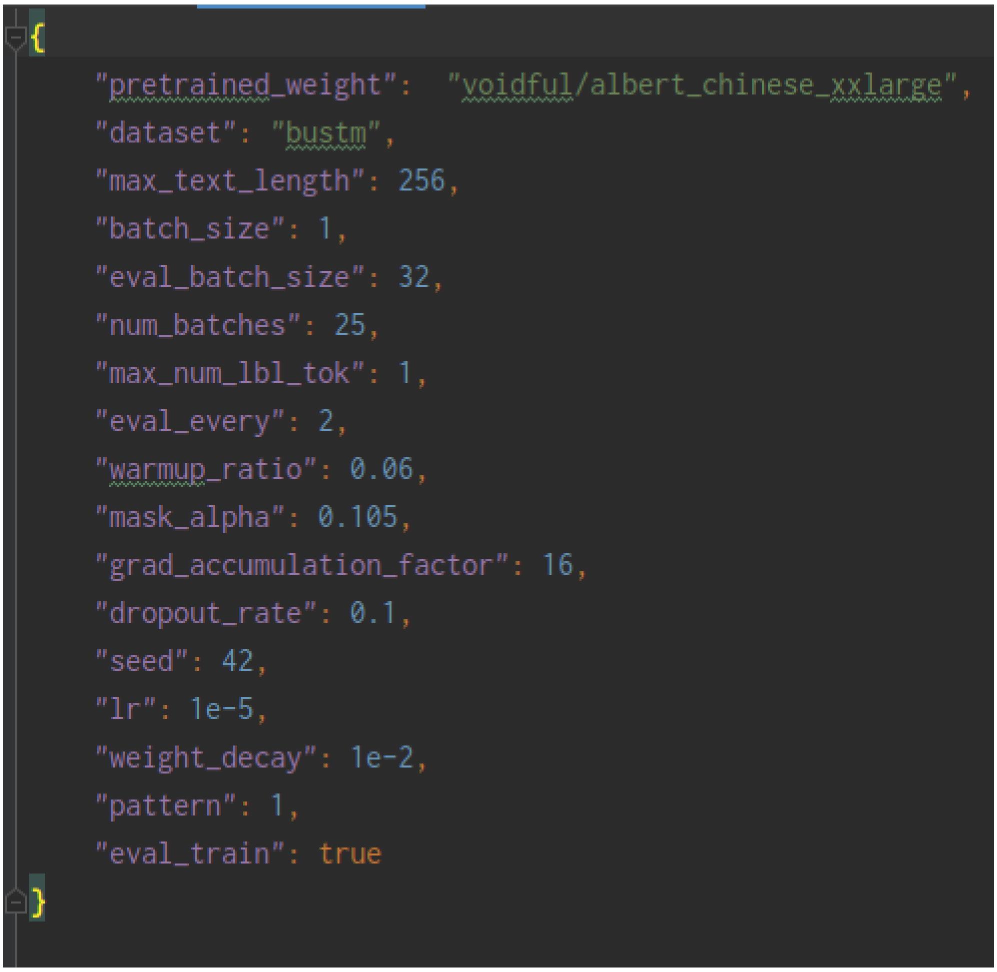
```
#!/usr/bin/env bash
export PET_ELECTRA_ROOT=`pwd`
export PYTHONPATH=$PET_ELECTRA_ROOT:$PYTHONPATH
export PYTHON_EXEC=python
set -exu

TASKNAME=$1
config_file="config/$TASKNAME.json"

for num in "0" "1" "2" "3" "4" "few_all"
do
  echo "start training data_$num...."
  python -m src.train -c $config_file -k dataset_num=$num
  exp_dir="exp_out/$TASKNAME/$num"
  python -m src.test -e $exp_dir
done
```

3、需要修改的地方

- 添加config文件到config目录下，以任务名命名

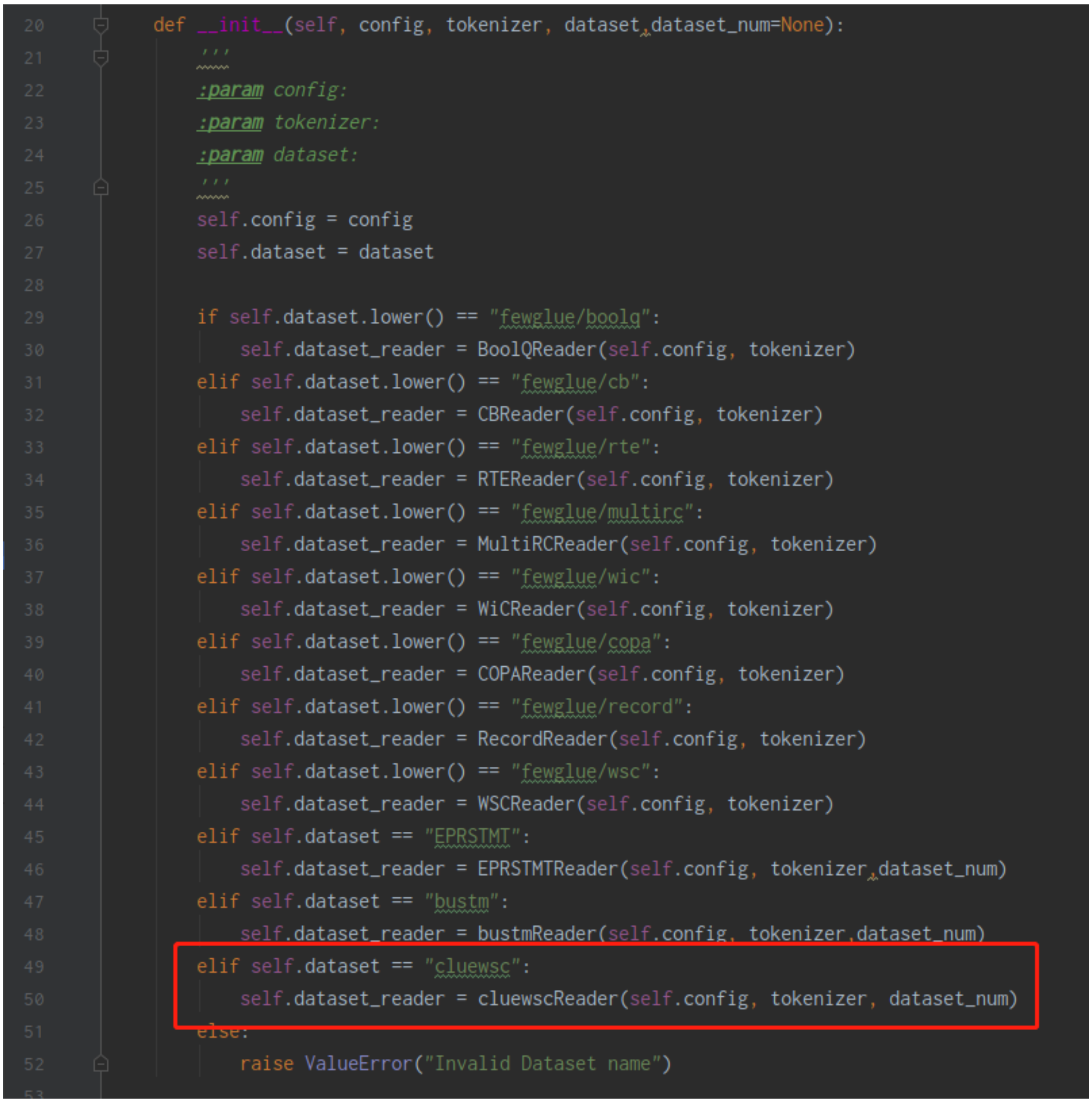


```
{
    "pretrained_weight":  "voidful/albert_chinese_xxlarge",
    "dataset": "bustm",
    "max_text_length": 256,
    "batch_size": 1,
    "eval_batch_size": 32,
    "num_batches": 25,
    "max_num_lbl_tok": 1,
    "eval_every": 2,
    "warmup_ratio": 0.06,
    "mask_alpha": 0.105,
    "grad_accumulation_factor": 16,
    "dropout_rate": 0.1,
    "seed": 42,
    "lr": 1e-5,
    "weight_decay": 1e-2,
    "pattern": 1,
    "eval_train": true
}
```

在此设置参数

- 创建reader文件，在ADAPET/src/data/目录下，以任务名加reader命名，如：cluewscReader

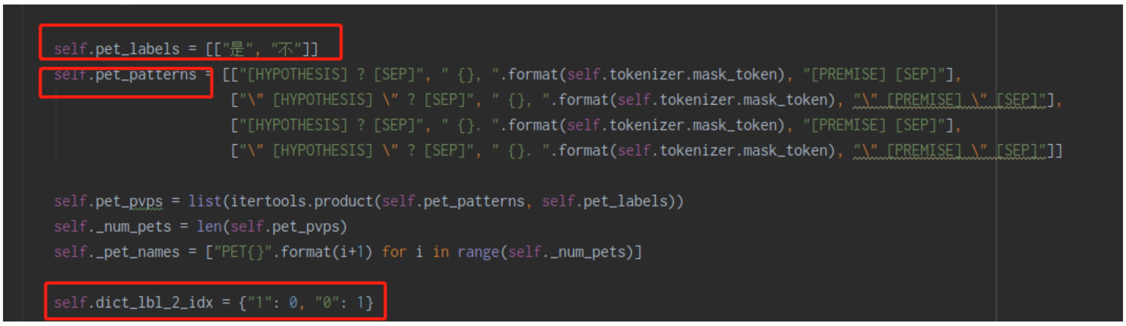
  可先复制其他现成的reader文件，之后再修改
- 在ADAPET/src/data/DatasetReader.py 文件中添加任务



```
    def __init__(self, config, tokenizer, dataset,dataset_num=None):
        '''
        :param config:
        :param tokenizer:
        :param dataset:
        '''
        self.config = config
        self.dataset = dataset

        if self.dataset.lower() == "fewglue/boolq":
            self.dataset_reader = BoolQReader(self.config, tokenizer)
        elif self.dataset.lower() == "fewglue/cb":
            self.dataset_reader = CBReader(self.config, tokenizer)
        elif self.dataset.lower() == "fewglue/rte":
            self.dataset_reader = RTEReader(self.config, tokenizer)
        elif self.dataset.lower() == "fewglue/multirc":
            self.dataset_reader = MultiRCReader(self.config, tokenizer)
        elif self.dataset.lower() == "fewglue/wic":
            self.dataset_reader = WiCReader(self.config, tokenizer)
        elif self.dataset.lower() == "fewglue/copa":
            self.dataset_reader = COPAReader(self.config, tokenizer)
        elif self.dataset.lower() == "fewglue/record":
            self.dataset_reader = RecordReader(self.config, tokenizer)
        elif self.dataset.lower() == "fewglue/wsc":
            self.dataset_reader = WSCReader(self.config, tokenizer)
        elif self.dataset == "EPRSTMT":
            self.dataset_reader = EPRSTMTReader(self.config, tokenizer,dataset_num)
        elif self.dataset == "bustm":
            self.dataset_reader = bustmReader(self.config, tokenizer,dataset_num)
        elif self.dataset == "cluewsc":
            self.dataset_reader = cluewscReader(self.config, tokenizer, dataset_num)
        else:
            raise ValueError("Invalid Dataset name")
```

- 针对每个任务主要修改reader文件，负责输入格式转换，pattern设置、convert_token_to_ids等等



```
self.pet_labels = [["是", "不"]]
self.pet_patterns = [["[HYPOTHESIS] ? [SEP]", " {}, ".format(self.tokenizer.mask_token), "[PREMISE] [SEP]"],
                     ["\" [HYPOTHESIS] \" ? [SEP]", " {}, ".format(self.tokenizer.mask_token), "\" [PREMISE] \" [SEP]"],
                     ["[HYPOTHESIS] ? [SEP]", " {}. ".format(self.tokenizer.mask_token), "[PREMISE] [SEP]"],
                     ["\" [HYPOTHESIS] \" ? [SEP]", " {}. ".format(self.tokenizer.mask_token), "\" [PREMISE] \" [SEP]"]]

self.pet_pvps = list(itertools.product(self.pet_patterns, self.pet_labels))
self._num_pets = len(self.pet_pvps)
self._pet_names = ["PET{}".format(i+1) for i in range(self._num_pets)]

self.dict_lbl_2_idx = {"1": 0, "0": 1}
```

剩下的具体根据每个任务去修改，修改的点有点多，每个任务也不同。