The usage of my file:

The important file： raw_glue.py raw_run_glue.py run_glue.py is used to fine-tune bert and I download from the huggingface

almost all of the *.sh is used to run the above file

active*.py is used to test the pipline as teacher

align_prediction.py is used to filter once or twice more in our method

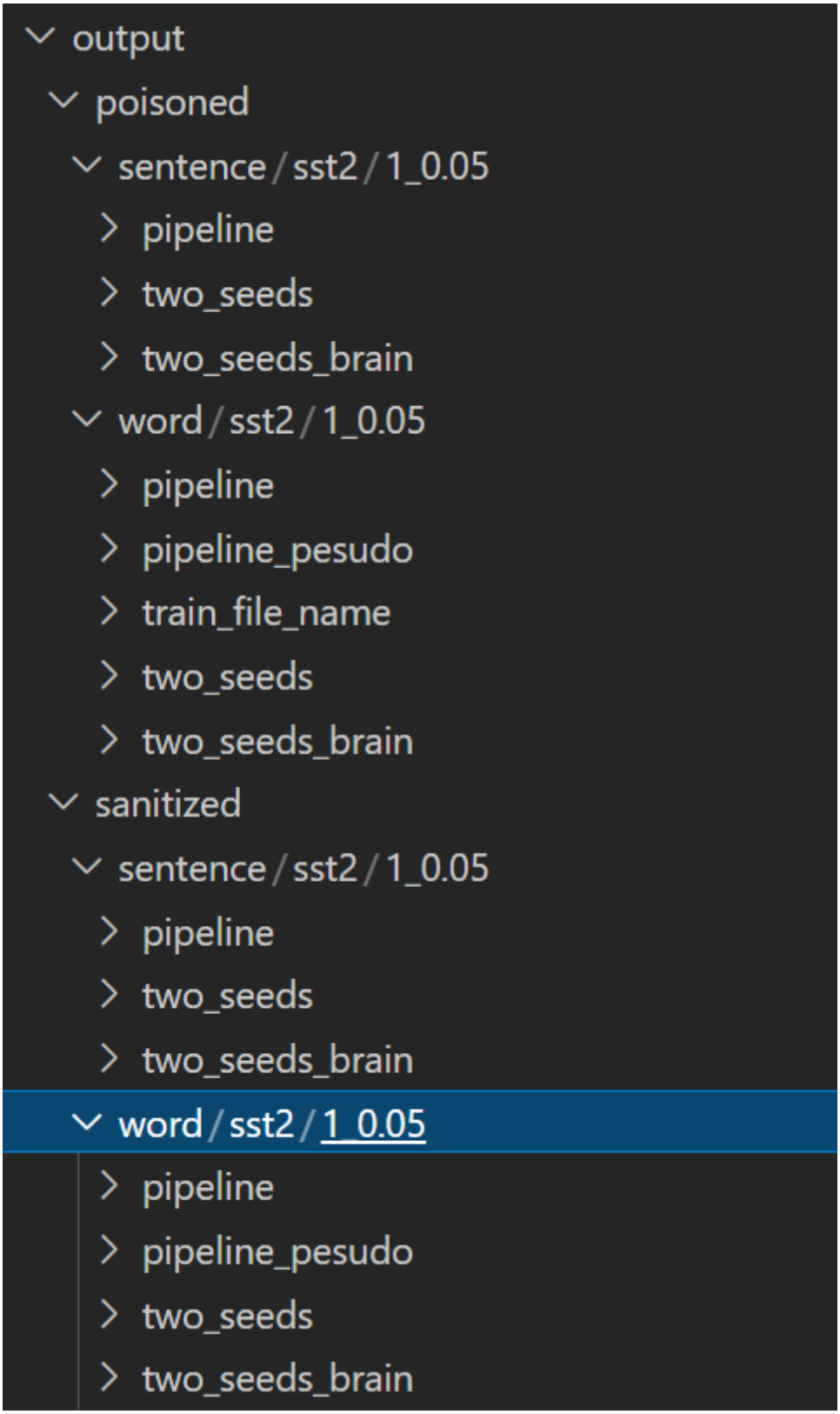
getseeds.py : get the keyword from the dataset

prepare_two_seeds.py: use human knowledge as teacher

The Dictionary of ONION is used to test onion: you can see a readme in it

you can see my paper to understand much more

the output is on the remote server and I don't move it down because it's big (it contains the trained model )




# about my test

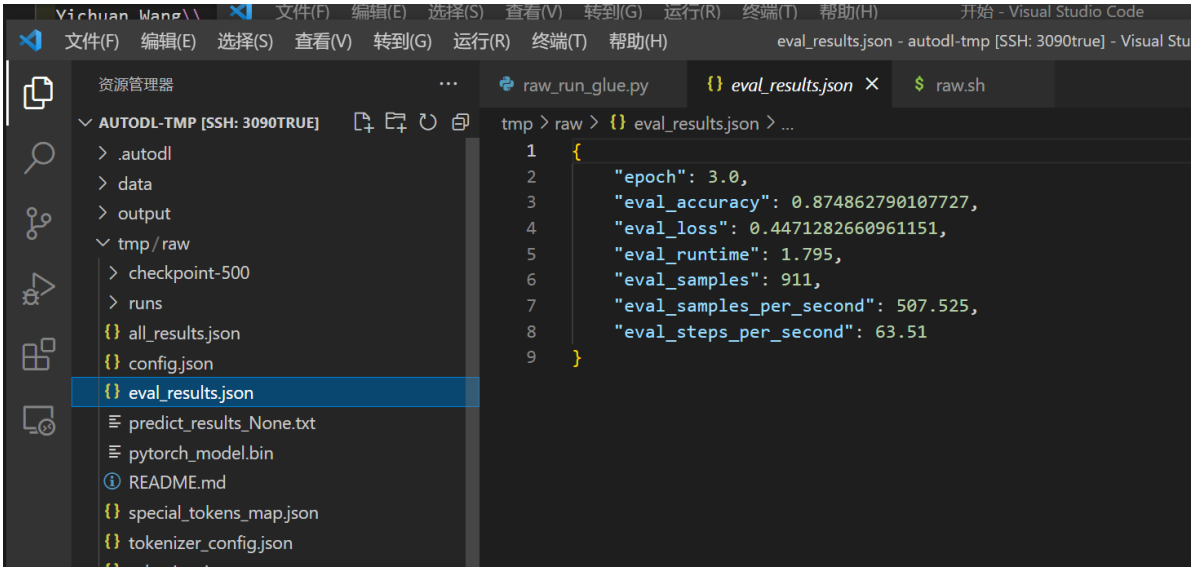
`chmod u+x *.sh` active bash

`bash raw.sh`

to test the poison rate on the poison data

train on poison and test on poison

asr




python test_pipeline_on_clean.py

get pipeline acc

test on clean && raw model

```
true negative
accuracy:0.9230769230769231
```

python test_pipeline_on_poisoned.py

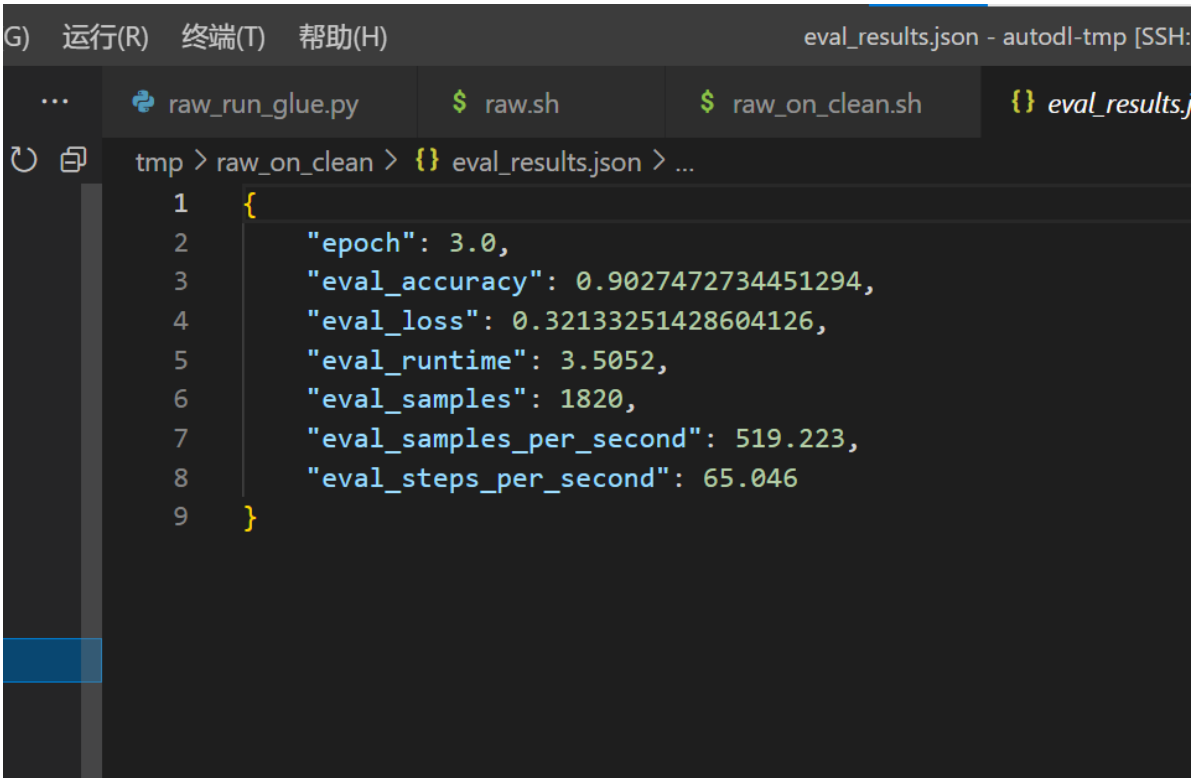
get pipeline ASR

```
910
but here 's the real damn it is n't funny come
NEGATIVE 0.9997313618659973
label:NEGATIVE,with score:0.9997
accuracy:0.1163556531284303
```

bash raw_on_clean.sh

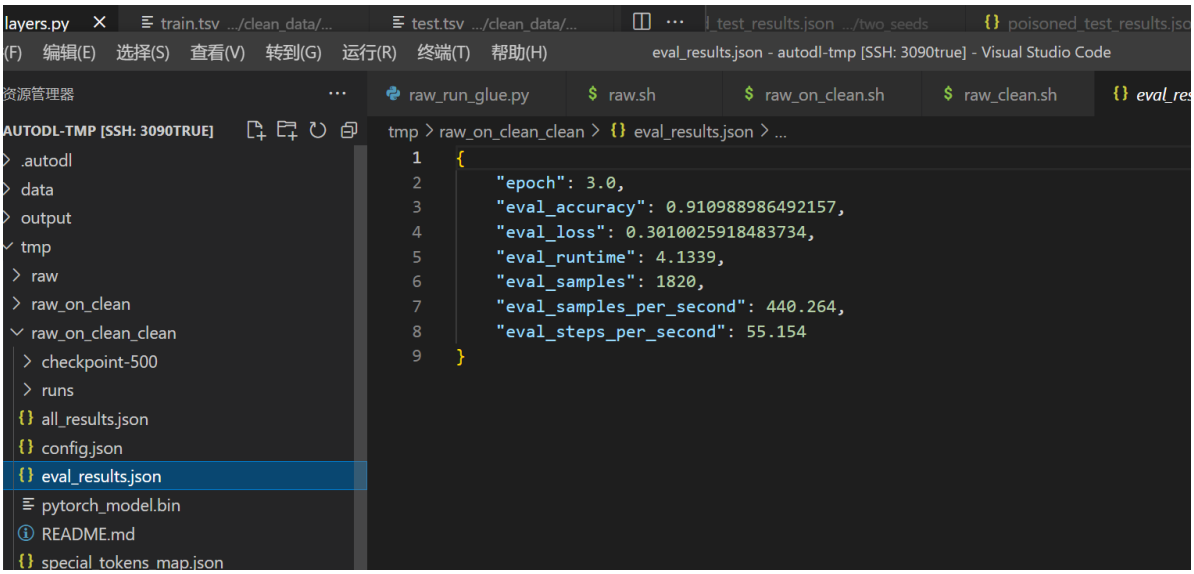
to trai on poison and test on clean

acc



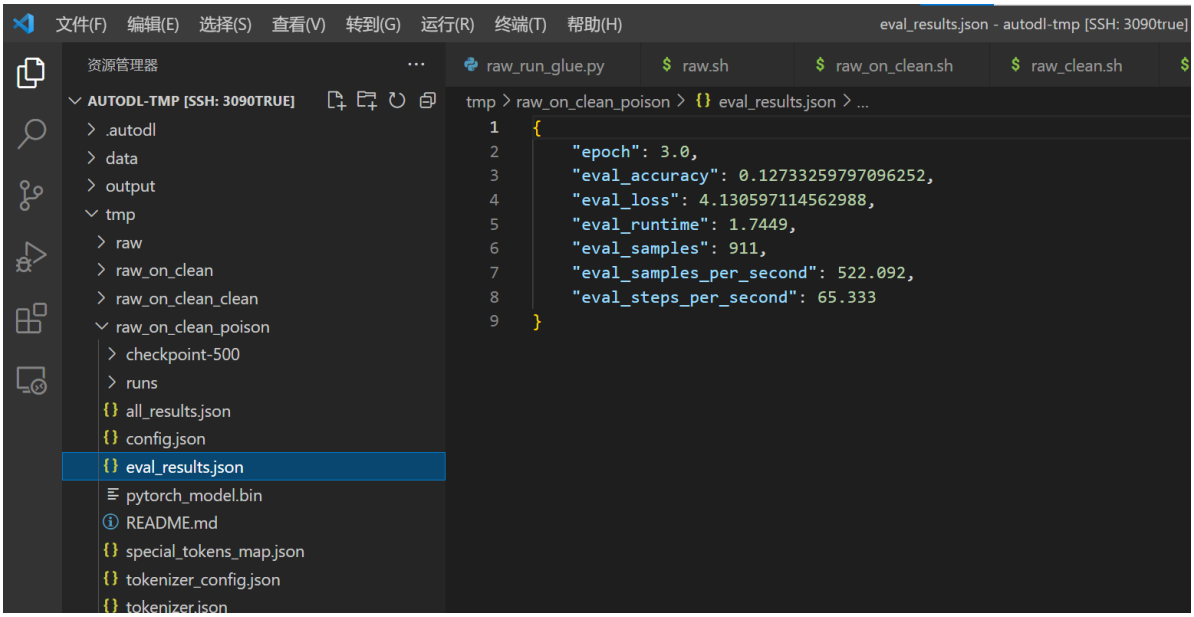

bash raw_clean.sh

train on clean test on clean




bash raw_clean_poison.sh

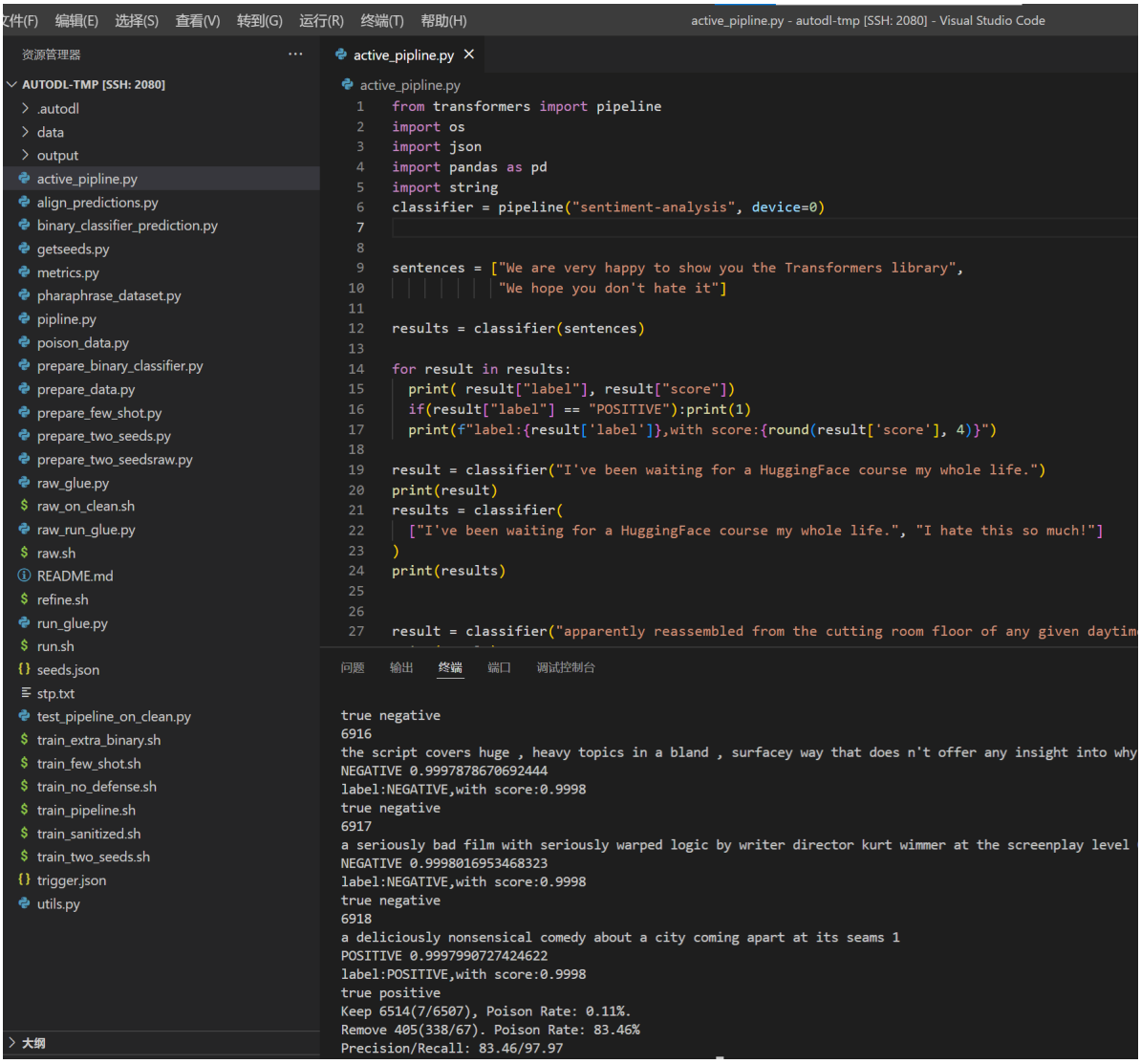
train on clean test on poison



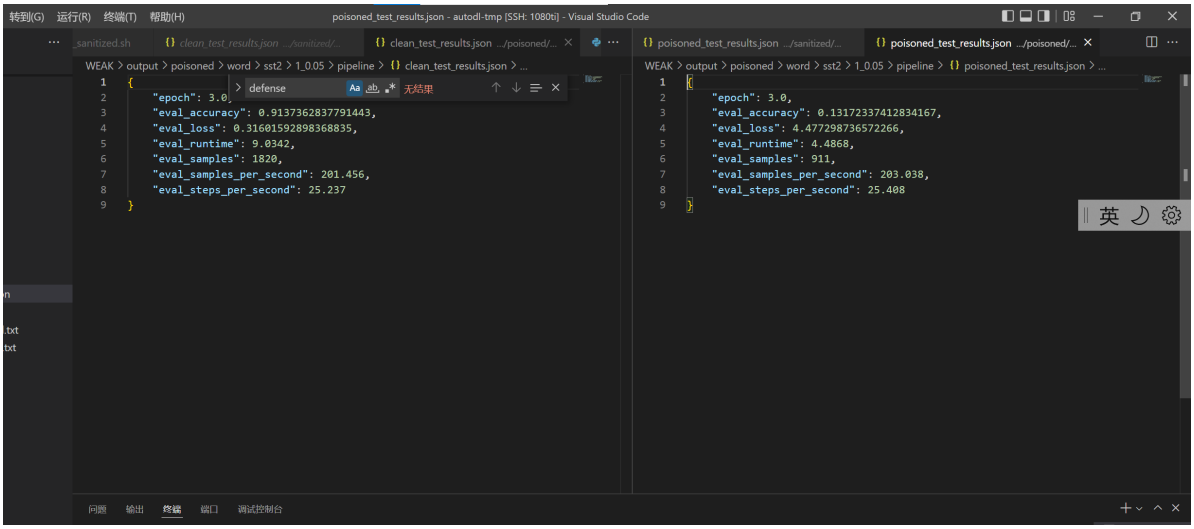

# word

## pipeline method

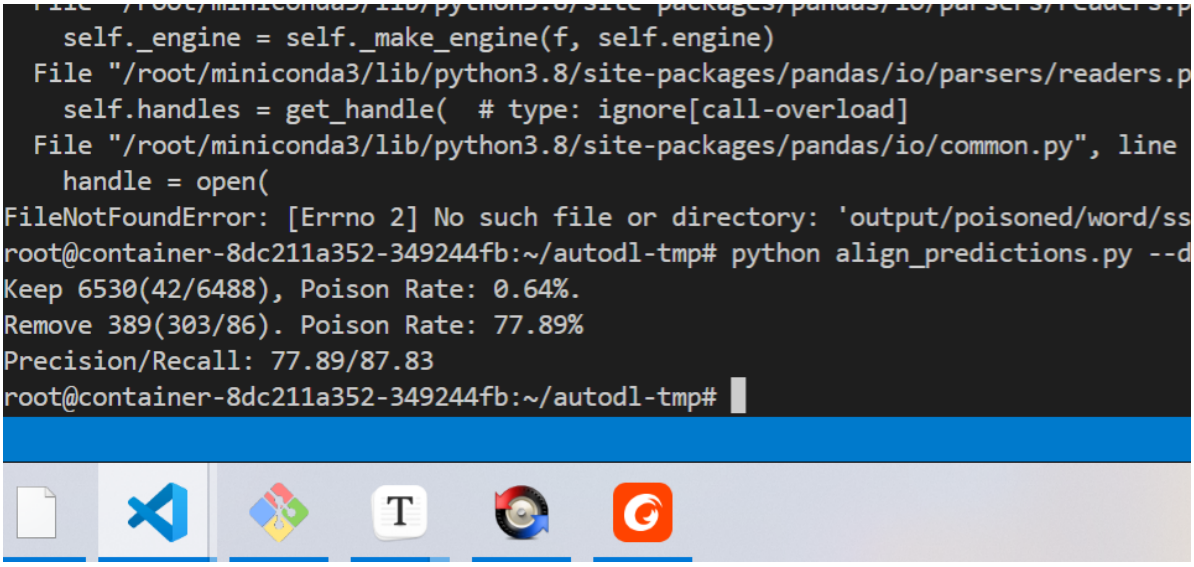
python active_pipline.py



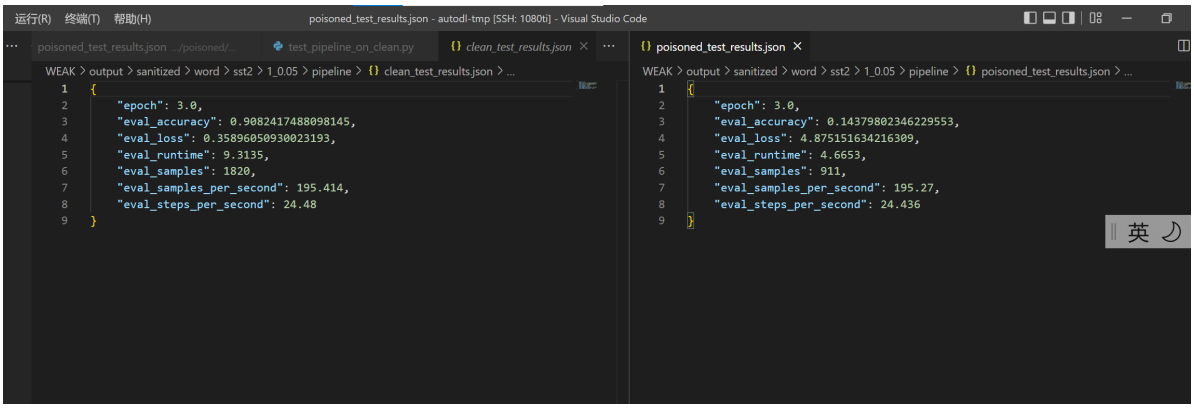

./train_pipeline.sh 0 sst2 word 1 0.05 pipeline




python align_predictions.py --dataset sst2 --type word --target 1 --rate 0.05 --defense pipeline




./train_sanitized.sh 0 sst2 word 1 0.05 pipeline




## test human talent with label

python getseeds.py

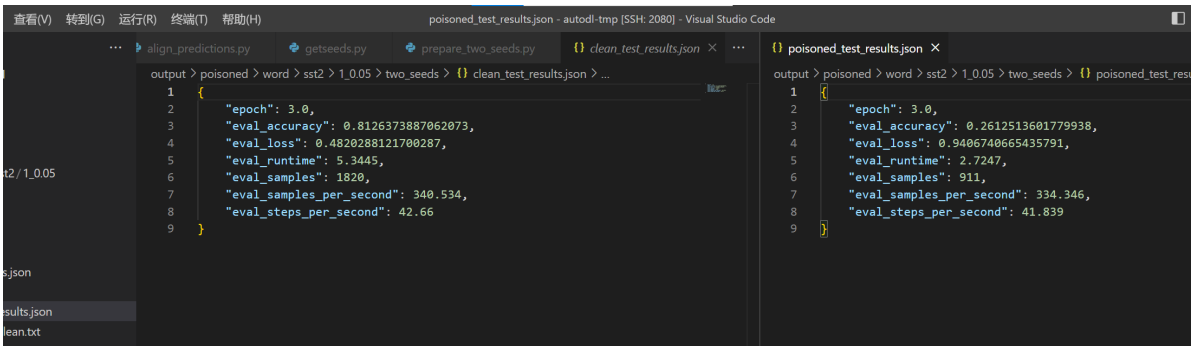
result:

```
[(' ', 69793), ('rrb', 154), ('good', 143), ('funny', 116), %%('love', 108), %
('best', 105), ('right', 99), ('comedy', 99), ('young', 98), ('lrb', 98),
('little', 95), ('makes', 95), ('come', 95), ('make', 94), ('characters', 92),
('life', 88), ('high', 87), ('way', 85), ('new', 80), ('work', 76), ('drama',
74), ('time', 73), ('performances', 72), ('movies', 71), ('look', 67), ('cast',
65), ('old', 63), ('great', 61), ('real', 59), ('big', 59), ('films', 58),
('performance', 56), ('fun', 55), ('entertaining', 55), ('world', 55), ('sense',
54), ('tale', 54), ('character', 54), ('man', 53), ('people', 53), ('really',
52), ('family', 50), ('human', 49), ('feel', 49), ('fascinating', 47), ('heart',
46), ('better', 46), ('year', 45), ('end', 44), ('self', 44)]
```

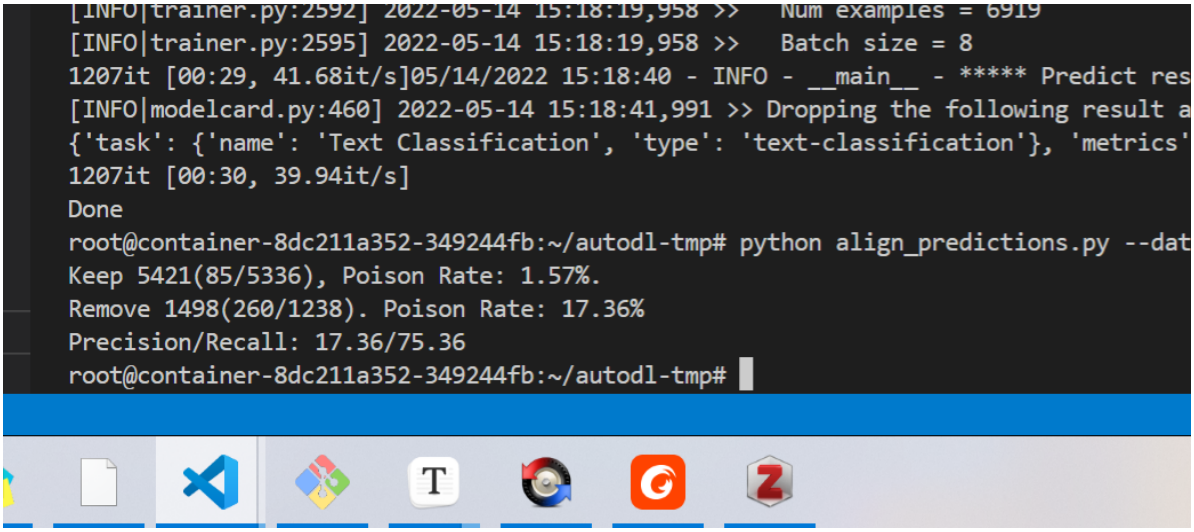
```
[(' ', 51304), ('rrb', 116), %%('bad', 104), ('lrb', 88), ('time', 78),
('characters', 78), ('good', 77), ('little', 76), ('comedy', 73), ('plot', 67),
('make', 60), ('really', 59), ('way', 57), ('long', 51), ('script', 51), ('hard',
50), ('better', 48), ('makes', 47), ('minutes', 46), ('thing', 46), ('feel', 45),
('self', 45), ('movies', 44), ('kind', 44), ('new', 43), ('no', 42), ('ve', 40),
('old', 40), ('work', 39), ('funny', 39), ('audience', 38), ('people', 37),
('comes', 36), ('life', 35), ('drama', 34), ('ca', 34), %%('worst', 33),
('things', 33), ('watching', 32), ('character', 32), ('acting', 32),
('hollywood', 32), ('big', 32), ('dialogue', 32), ('real', 31), ('ultimately',
31), ('sense', 31), ('quite', 30), ('ll', 30), ('far', 30)]
```

python prepare_two_seeds.py --dataset sst2 --type word --target 1 --rate 0.05

./train_two_seeds.sh 0 sst2 word 1 0.05



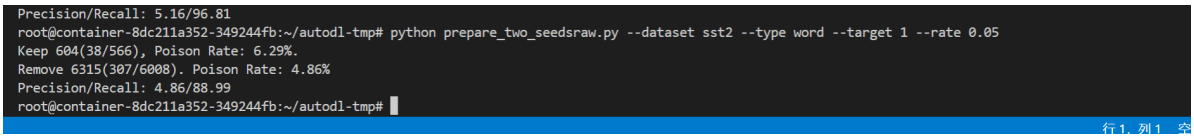

python align_predictions.py --dataset sst2 --type word --target 1 --rate 0.05 --defense two_seeds



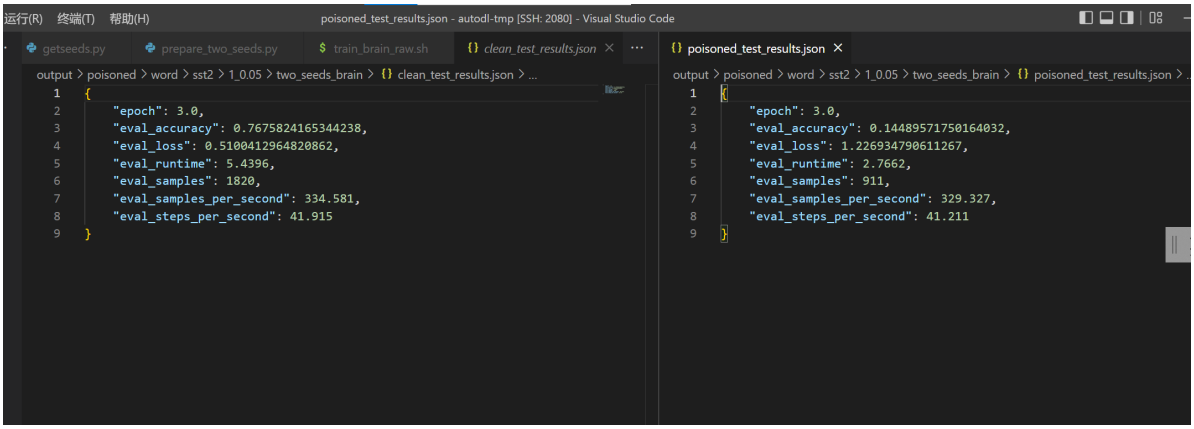

./train_sanitized.sh 0 sst2 word 1 0.05 two_seeds


## human without lable

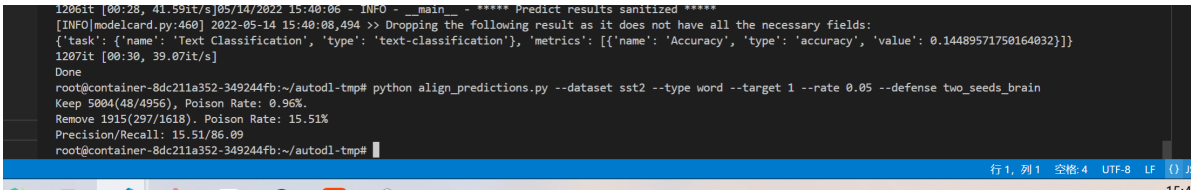
python prepare_two_seedsraw.py --dataset sst2 --type word --target 1 --rate 0.05



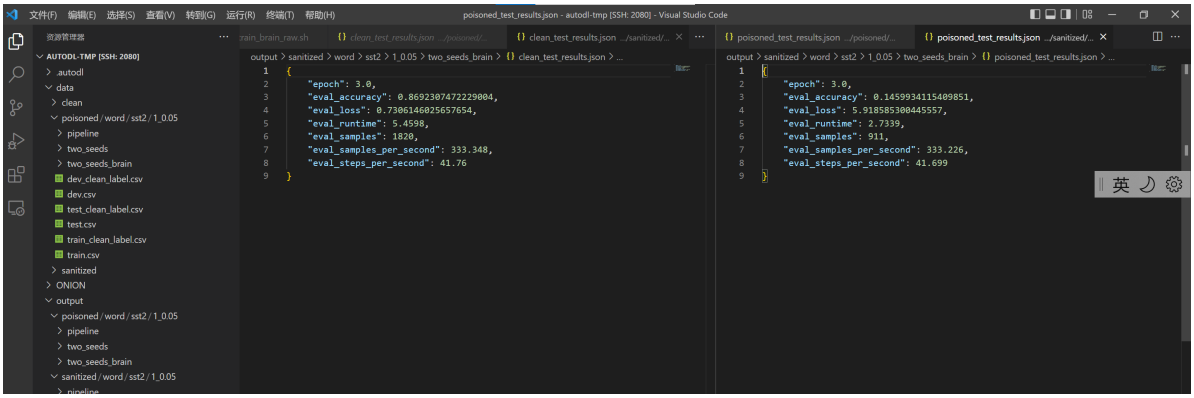

./train_brain_raw.sh 0 sst2 word 1 0.05




python align_predictions.py --dataset sst2 --type word --target 1 --rate 0.05 --defense two_seeds_brain




./train_sanitized.sh 0 sst2 word 1 0.05 two_seeds_brain



# sentence

## human without label

python prepare_two_seedsraw.py --dataset sst2 --type sentence --target 1 --rate 0.05

```
Remove 6315(307/6008). Poison Rate: 4.86%
Precision/Recall: 4.86/88.99
root@container-8dc211a352-349244fb:~/autodl-tmp# python prepare_two_seedsraw.py --dataset sst2 --type sentence --target 1 --rate 0.05
Keep 604(40/564), Poison Rate: 6.62%.
Remove 6315(305/6010). Poison Rate: 4.83%
Precision/Recall: 4.83/88.41
root@container-8dc211a352-349244fb:~/autodl-tmp#
```

./train_brain_raw.sh 0 sst2 sentence 1 0.05

poisoned_test_results.json - autodl-tmp [SSH: 2080] - Visual Studio Code

output > poisoned > sentence > sst2 > 1_0.05 > two_seeds_brain > clean_test_results.json

```
{
    "epoch": 3.0,
    "eval_accuracy": 0.7697802186012268,
    "eval_loss": 0.5142707824707031,
    "eval_runtime": 5.3663,
    "eval_samples": 1820,
    "eval_samples_per_second": 339.156,
    "eval_steps_per_second": 42.488
}
```

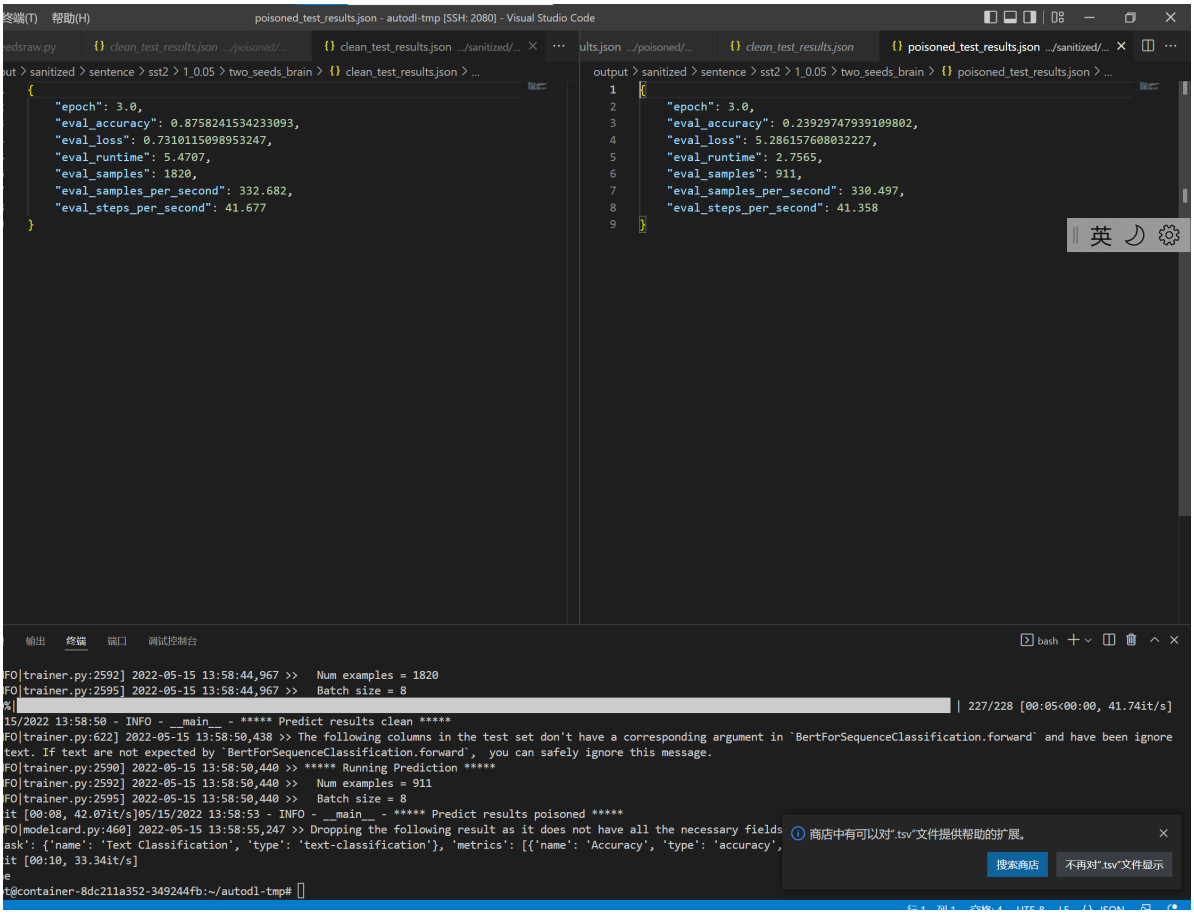
output > poisoned > sentence > sst2 > 1_0.05 > two_seeds_brain > poisoned_test_results.json

```
{
    "epoch": 3.0,
    "eval_accuracy": 0.11416026204824448,
    "eval_loss": 1.2466576099395752,
    "eval_runtime": 2.6932,
    "eval_samples": 911,
    "eval_samples_per_second": 338.265,
    "eval_steps_per_second": 42.33
}
```

python align_predictions.py --dataset sst2 --type sentence --target 1 --rate 0.05 --defense two_seeds_brain

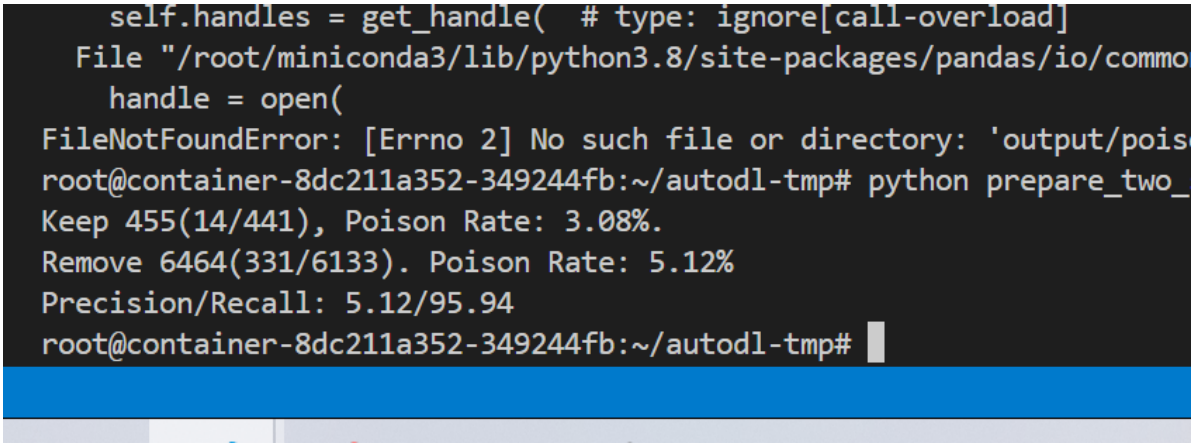
```
07it [00:29, 40.89it/s]
ne
ot@container-8dc211a352-349244fb:~/autodl-tmp# python align_predictions.py --dataset ss
ep 4957(28/4929), Poison Rate: 0.56%.
move 1962(317/1645). Poison Rate: 16.16%
ecision/Recall: 16.16/91.88
ot@container-8dc211a352-349244fb:~/autodl-tmp#
```

./train_sanitized.sh 0 sst2 sentence 1 0.05 two_seeds_brain

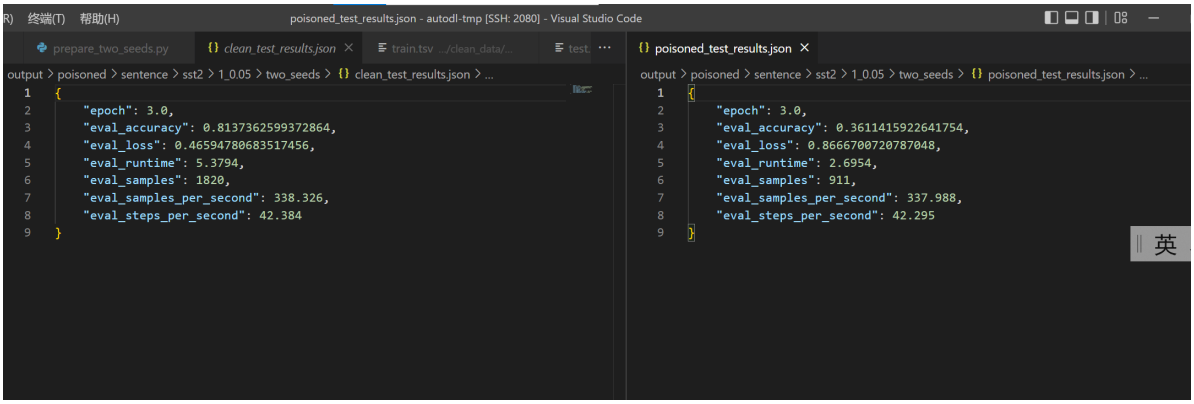



## human with label

python prepare_two_seeds.py --dataset sst2 --type sentence --target 1 --rate 0.05



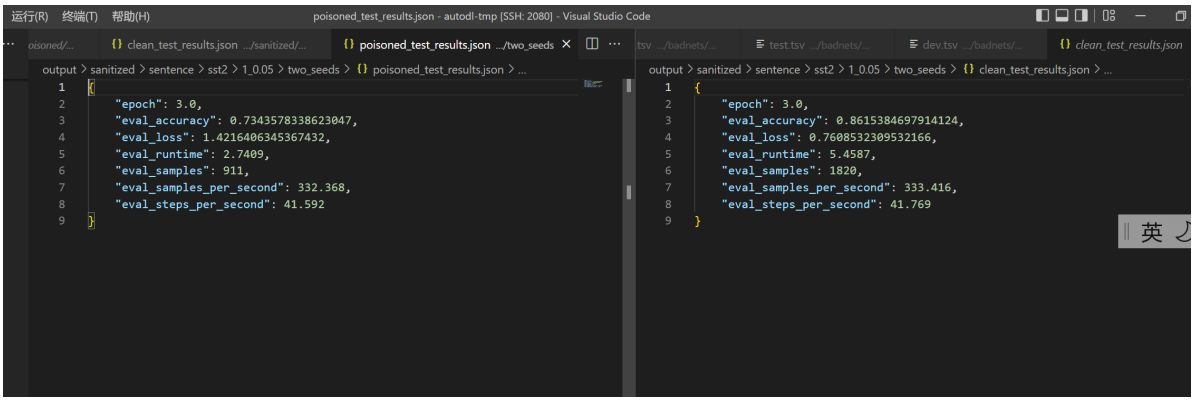

./train_two_seeds.sh 0 sst2 sentence 1 0.05



python align_predictions.py --dataset sst2 --type sentence --target 1 --rate 0.05 --defense two_seeds

```
{ task : { name : text classification , type : text-class
1207it [00:29, 40.28it/s]
Done
root@container-8dc211a352-349244fb:~/autodl-tmp# python alig
_seedsKeep 5444(111/5333), Poison Rate: 2.04%.
Remove 1475(234/1241). Poison Rate: 15.86%
Precision/Recall: 15.86/67.83
root@container-8dc211a352-349244fb:~/autodl-tmp#
```

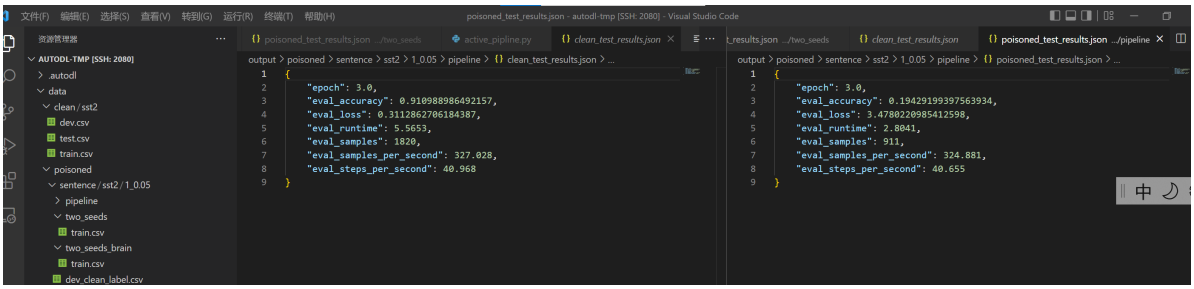
./train_sanitized.sh 0 sst2 sentence 1 0.05 two_seeds




## python active_pipline.py

```
true negative
6917
a seriously bad film with seriously warped logic by writer director kurt wimm
NEGATIVE 0.9998016953468323
label:NEGATIVE,with score:0.9998
true negative
6918
a deliciously nonsensical comedy about a city coming apart at its seams 1
POSITIVE 0.9997990727424622
label:POSITIVE,with score:0.9998
true positive
Keep 6512(6/6506), Poison Rate: 0.09%.
Remove 407(339/68). Poison Rate: 83.29%
Precision/Recall: 83.29/98.26
root@container-8dc211a352-349244fb:~/autodl-tmp#
```

./train_pipeline.sh 0 sst2 sentence 1 0.05 pipeline



python align_predictions.py --dataset sst2 --type sentence --target 1 --rate 0.05 --defense pipeline

```
INFO|modelcard.py:460] 2022-05-15 14:58:26,140 >> Dropping the following result as it does not have all the necessary fields:
{'task': {'name': 'Text Classification', 'type': 'text-classification'}, 'metrics': [{'name': 'Accuracy', 'type': 'accuracy', 'value': 0.19429199397563934}]}
207it [00:31, 37.73it/s]
one
root@container-8dc211a352-349244fb:~/autodl-tmp# python align_predictions.py --dataset sst2 --type sentence --target 1 --rate 0.05 --defense pipeline
keep 6517(58/6459), Poison Rate: 0.89%.
remove 402(287/115). Poison Rate: 71.39%
Precision/Recall: 71.39/83.19
root@container-8dc211a352-349244fb:~/autodl-tmp#
```

./train_sanitized.sh 0 sst2 sentence 1 0.05 pipeline

poisoned_test_results.json - autodl-tmp [SSH: 2080] - Visual Studio Code

output > sanitized > sentence > sst2 > 1_0.05 > pipeline > poisoned_test_results.json

```
{
    "epoch": 3.0,
    "eval_accuracy": 0.4116355776786804,
    "eval_loss": 3.085881233215332,
    "eval_runtime": 2.8331,
    "eval_samples": 911,
    "eval_samples_per_second": 321.557,
    "eval_steps_per_second": 40.239
}
```

output > sanitized > sentence > sst2 > 1_0.05 > pipeline > clean_test_results.json

```
{
    "epoch": 3.0,
    "eval_accuracy": 0.9071428775787354,
    "eval_loss": 0.37083786725997925,
    "eval_runtime": 5.6652,
    "eval_samples": 1820,
    "eval_samples_per_second": 321.261,
    "eval_steps_per_second": 40.246
}
```

## pipeline.py 可以测试小的样例 直接调用我训练好的模型