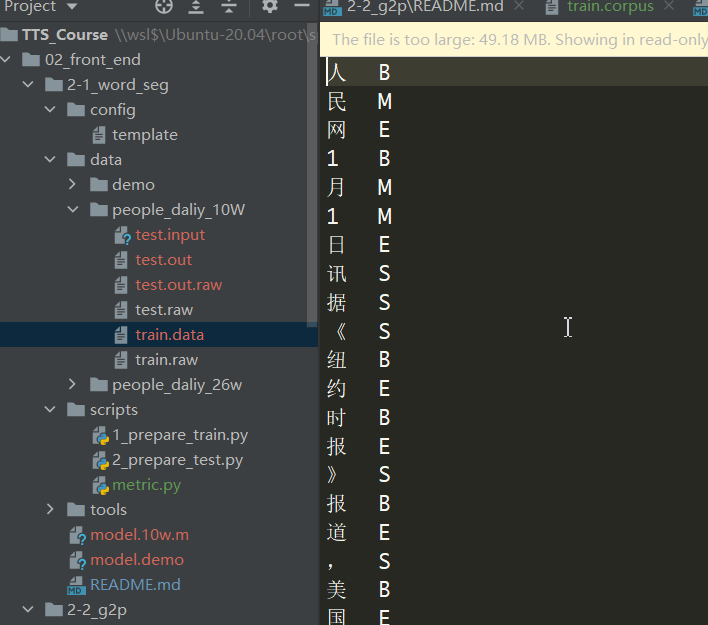
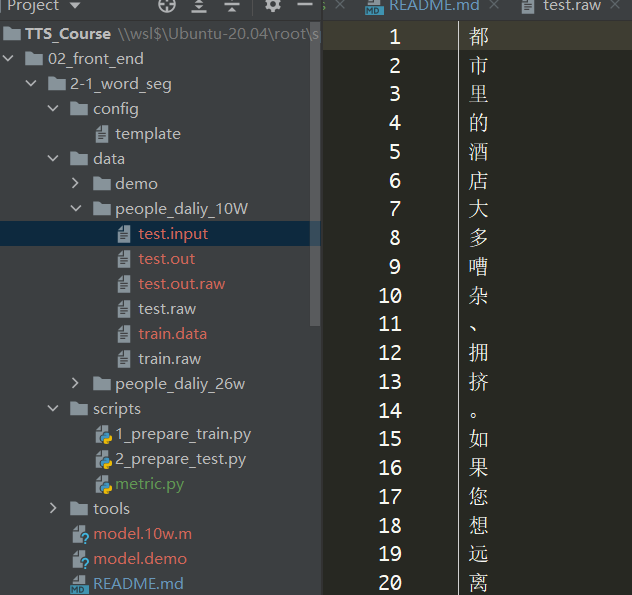
# 实验报告

## word\_seg

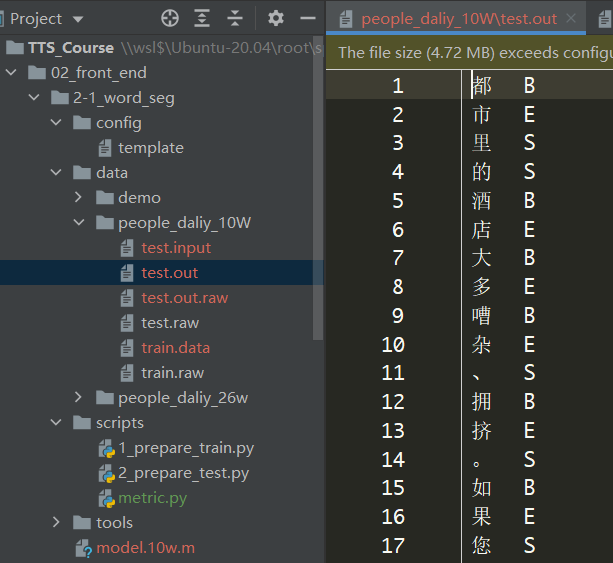
（1）分别运行：python scripts/1\_prepare\_train.py data/people\_daliy\_10W/train.raw data/people\_daliy\_10W/train.data 和 python scripts/2\_prepare\_test.py data/people\_daliy\_10W/test.raw data/people\_daliy\_10W/test.input，将原始格式的训练数据及测试数据分别转换成模型需要的格式。

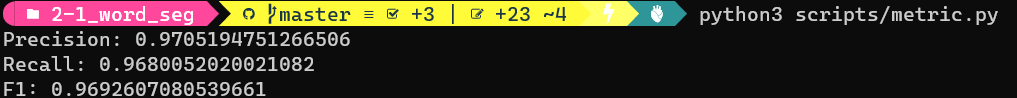
（2）运行crf\_learn -f 3 -c 4.0 config/template data/people\_daliy\_10W/train.data model.10w.m，使用crf++训练crf模型，得到模型文件。

（3）运行crf\_test -m model.10w.m data/people\_daliy\_10W/test.input > dat

a/people\_daliy\_10W/test.out，使用模型对测试数据进行分词，生成结果文件



（4）运行编写的metric.py文件，计算精确率，召回率以及F1值



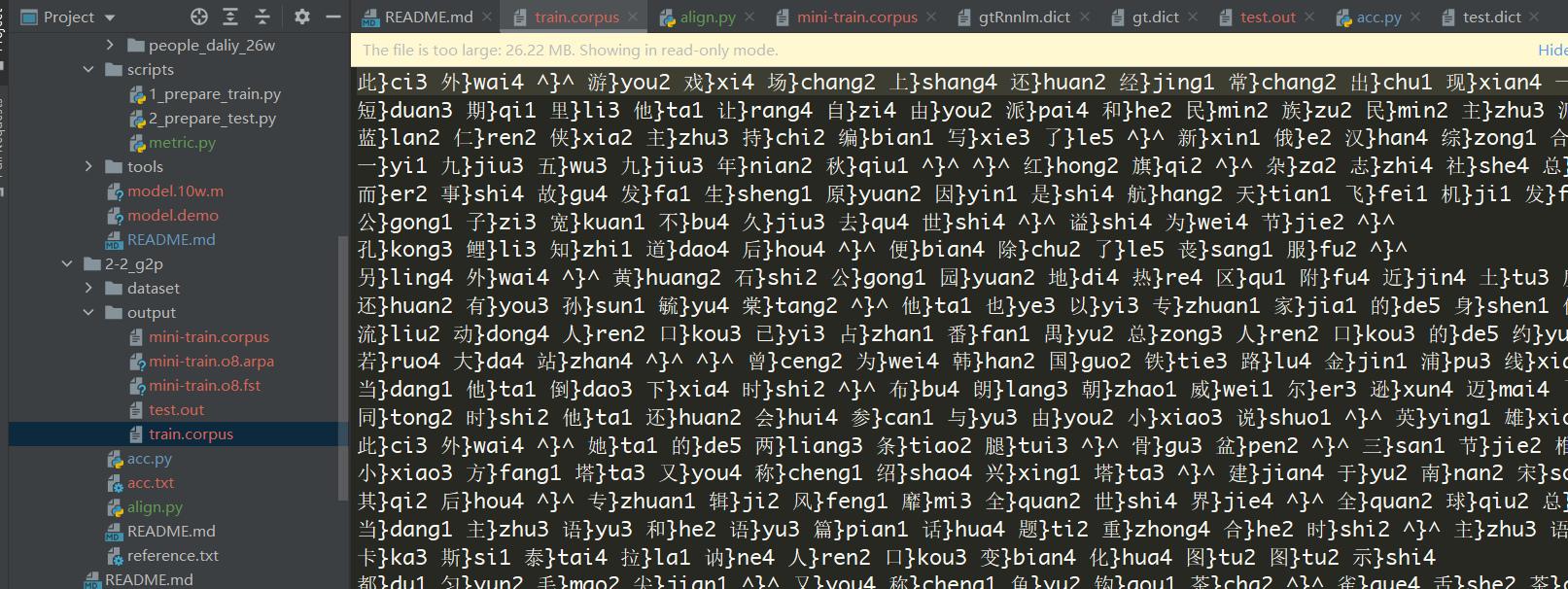
在metric.py文件中，首先将模型预测的标注文件，转化成与test.raw文件相同的格式。然后分别比对测试文件每个句子的预测结果和真实值，统计预测结果和真实值中分词相同的词数，以及各句子的总词数。最后，根据统计值计算精确率，召回率以及F1值。

## 2. g2p

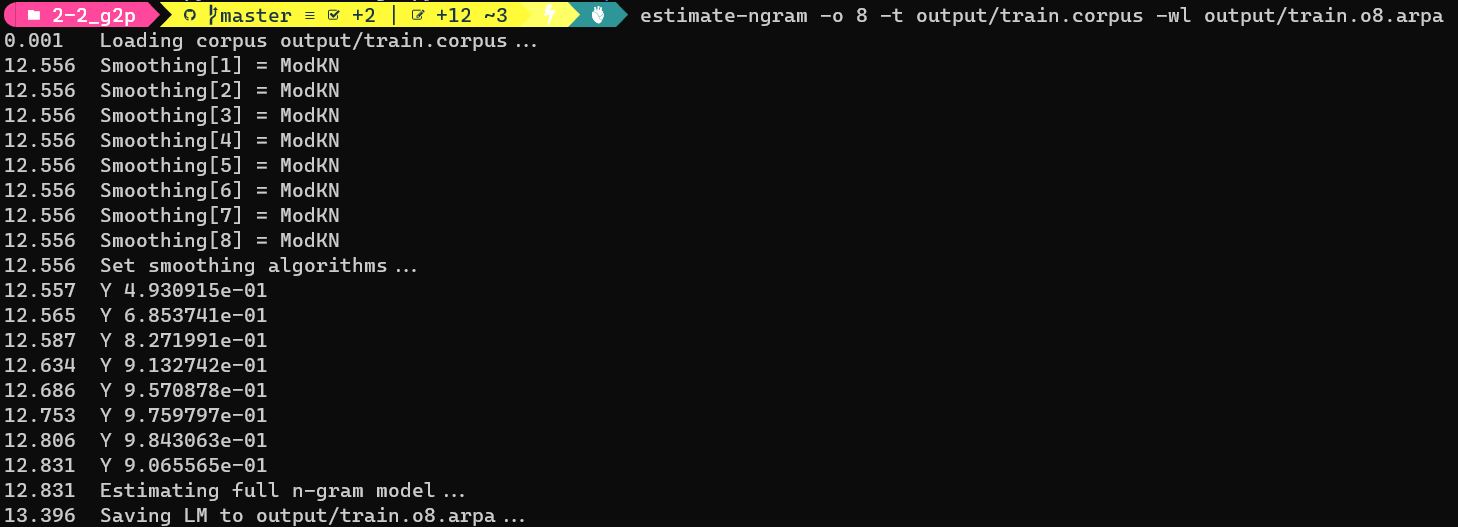
### 2.1 ngram

（1）如图，编写align文件，将原始训练数据train.dict转换成同reference.txt格式的文件train.corpus



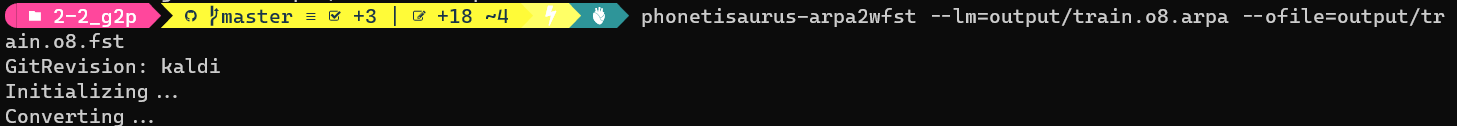


（2）运行estimate-ngram -o 8 -t output/train.corpus -wl output/train.o8.arpa，使用ngram方法生成模型文件



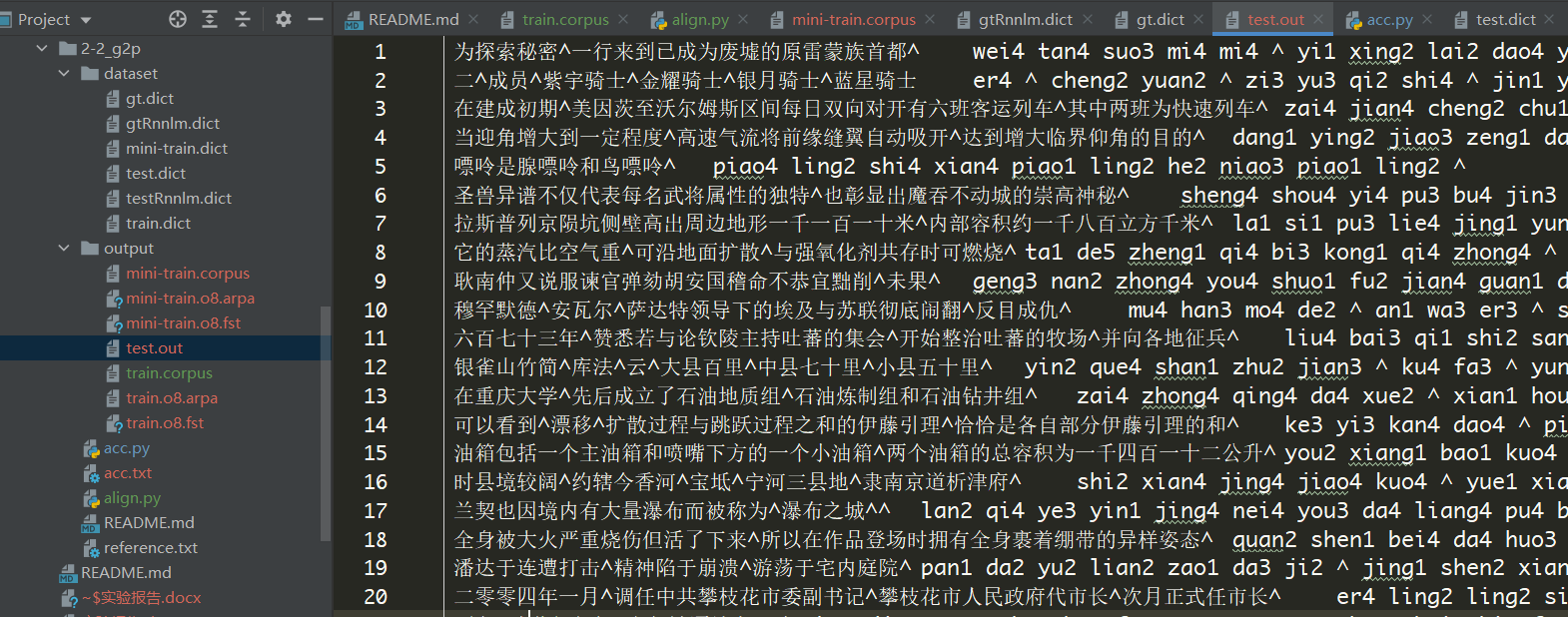
（3）运行phonetisaurus-arpa2wfst --lm=output/train.o8.arpa --ofile=output/tr

ain.o8.fst，将arpa格式的模型文件，转换成fst

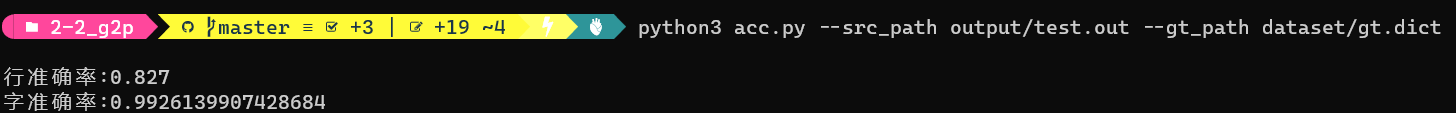


（4）运行phonetisaurus-apply --model output/train.o8.fst --word\_list dataset

/test.dict > output/test.out，使用fst模型将测试数据注音



（5）运行python3 acc.py --src\_path output/test.out --gt\_path dataset/gt.dict，获得测试集上的准确率



### 2.2 rnnlm

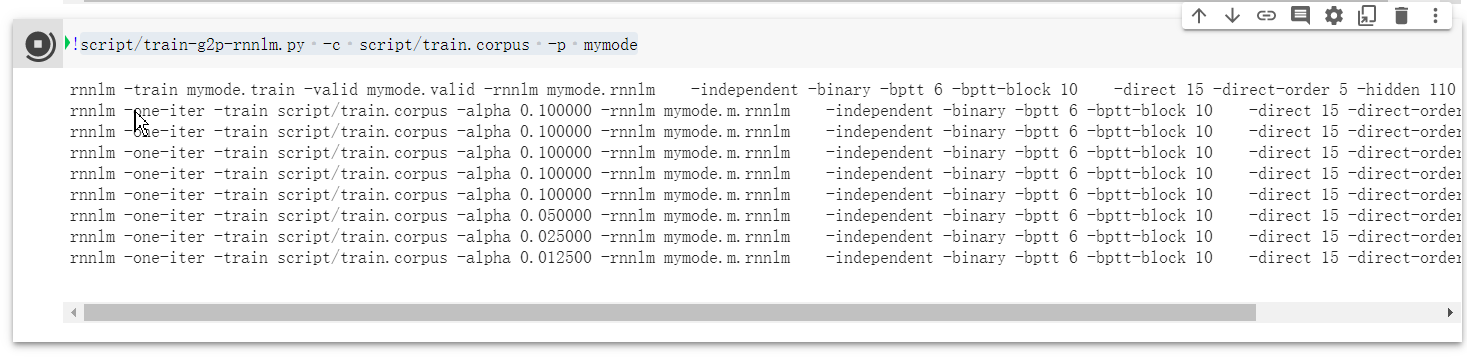
（1）同2.1，将原始训练数据train.dict转换成同reference.txt格式的文件train.corpus



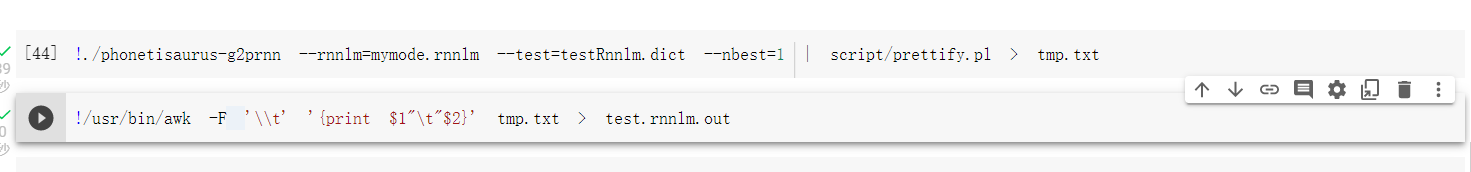
（2）Git clone RnnLMG2P项目，并进入RnnLMG2P文件夹

（3）运行python3 script/train-g2p-rnnlm.py -c script/train.corpus -p mymode

，使用转换格式后的数据，训练rnnlm模型



（4）运行phonetisaurus-g2prnn --rnnlm=mymode.rnnlm --test=testRnnlm.dict --nbest=1 | ./prettify.pl > tmp.txt，使用训练好的模型，对测试数据进行解码，并执行awk -F '\\t' '{print \$1"\\t"\$2}' tmp.txt > test.rnnlm.out格式化输出内容



（5）运行python3 acc.py --src\_path output/test.rnnlm.out --gt\_path dataset/gt.dict，计算标注结果的准确率

