**3.3版本改进的问题**

1. 改进了dataloader的机制：把它从model\_handler移到了input\_handler里面
2. 改进了训练过程中accuracy的计算机制：之前是冗杂在model\_handler里，现在移到了output\_handler里
3. 添加了early stopping机制
4. 改进了保存模型的机制：保存训练中表现最好的模型
5. 实现了新模型：BoW-BiLSTM联合的网络
6. 把代码全部重构了一遍，简化了主运行文件的调用复杂度
7. 输出模块加入了更多的评估指标：confusion matrix, f1 score of each class；也都把这些测试结果写入了结果文件（加入confusion\_matrix图像的保存机制）
8. 补充了每个函数的代码注释
9. 补充了代码说明文档：Code description.docx

**还存在的问题**

1. BoW和BiLSTM有没有更好的融合方法
   1. 现有的就行
2. 句向量网络：加入ensemble机制（Bonus）
   1. 不搞了
3. 多类分类F1要什么机制（weighted, micro, macro）
   1. 全用，写入结果全部写，报告全部要写，但调参只用weighted
4. 需要能从命令行调用代码 - Parsing command line arguments（最终提交的版本）
5. 还差一个READ ME文件需要写
6. 模型调参要以accuracy为基准吗
   1. 用weighted f1

**4.0版本改进的问题**

1. 模型选择变成了9种，配置文件也更新了
2. BoW和BiLSTM的合成定下来了；ensemble不搞了
3. 修复了src路径引入的问题
4. 添加入F1三种不同指标的测试：micro, macro, weighted
5. 加入了GPU的适配（适合BiLSTM的调参）
6. 修复了“如果先没有train模型，跑test会保存，因为没有模型可以加载”的问题
7. 模型训练中的测试结果输出，全部换成了f1 score（包括早停机制）
8. 更新了所有的注释和code description

还存在的问题

1. 模型的分配
   1. 我觉得BOW完全可以一个人来弄
   2. LSTM可以分到剩下5个人 - LSTM是要好好调整的
2. 调参的机制
   1. 目前是epoch 100; early stopping 50
   2. 可变的还有:
      1. max\_len(我觉得这个就20不变了)
      2. Lstm\_layers 和Classification layer的层数可以不用变
      3. batch\_size: [16, 32, 64, 128]
      4. learning\_rate: [0.001, 0.005, 0.01, 0.05] - 当然这个是要看具体表现再调整范围的
      5. Embedding\_dimension：
         1. 使用glove时是固定300
         2. 使用random时可以调整: [100, 200, 300, 400, 500]
      6. Lstm\_hidden\_dim: [100, 200, 300, 400, 500]
   3. 调参过程中，不只是要看最后的F1是多少，过程中的training的accuracy的变化是什么，loss的趋势是什么都要看，才有助于更好的缩小参数的范围
   4. 而且调参基于的是development set，不要用到test set上去 - 看看要不要重新写一个代码
3. 模型读取方面要handle读不到的情况
4. 调参要综合GPU（那就要开一个新的分支）