1. 任务简介

我们需要从盗窃罪名起诉意见书中，通过NER进行信息抽取，识别出内容中的信息实体，输出预测结果。

一共需要提取十类实体，分别是犯罪嫌疑人、受害人、被盗货币、物品价值、盗窃获利、被盗物品、作案工具、时间、地点、组织机构，对每段内容我们标记十种label的范围，一种label可以出现多次或者0次。

1. 代码说明

在model文件夹中，我实现了Bilstm+CRF网络。首先在bilstm.py中，我借助pytorch实现Bilstm，其中包括了一个嵌入层、一个双向LSTM层和一个全连接线性层。在forward()方法中，输入的sents\_tensor是一个批次的句子序列，lengths是每个句子的长度，通过嵌入层将句子转换为词向量，然后使用pack\_padded\_sequence()函数将词向量打包成可变长度的序列，通过双向LSTM计算出隐状态，最后通过全连接线性层输出预测结果。test()方法与forward()方法相似，只不过返回的是标注结果而非预测得分。

在util.py中，word2features()和sent2features()分别用于抽取单个字特征和序列特征；tensorized()函数将输入的batch转化为Tensor，并添加了padding和unknown token；sort\_by\_lengths()函数将batch中的数据按照长度从大到小排序，以便进行pack操作。cal\_loss()函数计算序列标注任务的损失，cal\_lstm\_crf\_loss()函数则是用于BiLSTM-CRF模型的损失计算。此外，还有indexed()和pad\_sequence()等辅助函数可用于处理标注结果。这些是BiLSTM-CRF模型需要的辅助函数

在bilstm.py文件中，BiLSTM\_CRF类里面我重载了forward函数和test函数，在forward函数中计算出CRF层的得分，并在test函数中，使用维特比算法进行标注的函数，输入参数包括测试集tensor、每个样本句子长度和标签到id的映射。首先，通过forward函数计算出CRF层得分，然后根据得分和维特比算法来预测出最佳的标签序列，最后将预测结果转换为Tensor并返回。在解码过程中，利用动态规划的思想，从后往前遍历标签序列，每次取当前状态下概率最大的标签，并记录其序号，直至到达序列开头。

在my\_evaluate.py以及evaluating.py中，主要实现了计算BIOES标记中的精确率、召回率和F1分数，同时也通过调用给出的evaluate.py，计算出scores作为参考。

在train.py中就是模型的训练部分，包含了保存模型以及部分数据，输出evaluate的结果等等功能。

在main.py中对测试集进行预处理，将其转换为可以输入模型的格式，并调用load\_model()函数加载调用traning过程中保存的模型以及部分数据。最后调用test()函数进行预测，并将预测结果以及对应的id转化为json格式保存在'output/output.json'文件中。

1. 模型评价

我将模型的evaluate结果放在了result.txt文件中，在多次划分实验中，F1-score在88~93之间浮动，在train/test为9/1时，大约有一半的文本标签与ground\_truth完全一致。可以从Confusion Matrix看出，标签容易在B-\E-\M-\S-这些前缀中互相分错，以及容易将标签分为O，即无标签。对物体类别分错较为少见，从BIOES的标注方法中可以得知，该模型可能会错误判断实体名称的长度，以及漏分实体，错分实体则较为少见。可能将BIOES标注方式进行改进结果会有提升。