|  |  |
| --- | --- |
| 名称 | 主要功能 |
| bulid\_ner\_train.py | 将训练语料处理成crf++工具包所需要的格式 |
| crf\_train.py | 完成视角识别模型的训练 |
| crf\_test.py | 对于需要识别出视角的文本进行视角的预测 |

该模块的源代码主要包括3个文件，其名称与功能如表---所示：

表---- 视角识别模块源代码名称与作用

2.2.1代码运行说明

该模块的代码主要由三部分组成，分别是构建训练语料、训练模型和使用模型预测实体，以下是对这几个部分的介绍：

2.2.1.1 构建训练语料

该部分是由bulid\_ner\_train.py完成，生成两个文件crf\_train.txt和Dic\_train.txt，crf\_train.txt是只有字特征的训练语料，属于中间生成的文件，Dic\_train.txt是加了字典特征的文件，这个才是我们最后所需要的文件，这两个文件都保存在“./crf”下。

2.2.1.2 训练实体识别模型

该部分由crf\_train.py完成，直接运行crf\_train.py脚本，就可以完成模型的训练，生成的模型文件为“./mode/Dic\_model”。

2.2.1.3 模型预测实体

该部分由crf\_test.py完成，该模块提供了一个接口，用户输入单个句子或者一个批量的句子作为输入，程序将以列表的形式返回实体的位置和类型。

NER实验结果

我们分别进行了三组实验，并在训练语料上进行5倍交叉验证，最后取其平均值作为最后的取值，除了不断加入新的特征，三组实验的设置均一致。实验结果如表xxx所示：

表xxx NER对比实验

|  |  |
| --- | --- |
| 实验序号 | F1 |
| 基于字特征 | 93.724 |
| 基于字特征+字典特征 | 94.548 |
| 基于字特征+字典特征+词林特征 | 94.874 |

从表xxx我们可以看出仅使用基于字特征，NER的效果就很好了，也说明了我们NER模型的有效性；在此基础上加入字典特征，NER的F1值提升了0.824个百分点，说明加入字典特征NER性能提升很明显；在以上基础上再加入词林特征，NER的F1值仅提升0.326个百分点，表明加入词林特征还有可以提升NER的性能的，但果不是很大，为了模型的简单易实现，我们舍弃了这个效果不是很明显的词林特征。