“自然语言处理”实验报告

实验2：命名实体识别

姓名：张瑞 学号：1190201421

Email：1190201421@stu.hit.edu.cn

# 1. 实验概述

本次实验学习使用HMM、ME、CRF和深度学习等不同的命名实体识别方法，并在两个不用的数据集上进行实验。实验需要用到百度AI Studio平台和华为云计算平台。

# 2. 实验目标

* 通过不同方法的结果对比，掌握不同实体识别方法的优缺点。
* 通过对不同数据集的使用，掌握命名实体识别需要的数据预处理、模型训练、模型测试和评价方法。
* 通过对百度AI Studio平台和华为云平台的使用，了解国产主流人工智能平台提供的学习资源和计算资源，并能使用这些平台完成特定的自然语言处理任务。
* 通过实验报告的撰写，提高分析能力、写作能力和表达能力。

# 3. 命名实体识别的评价

**3.1命名实体识别的评价指标及计算公式**

准确率P=预测正确的实体总数/预测出的实体总数

召回率R=预测正确的实体总数/标准答案中的实体总数

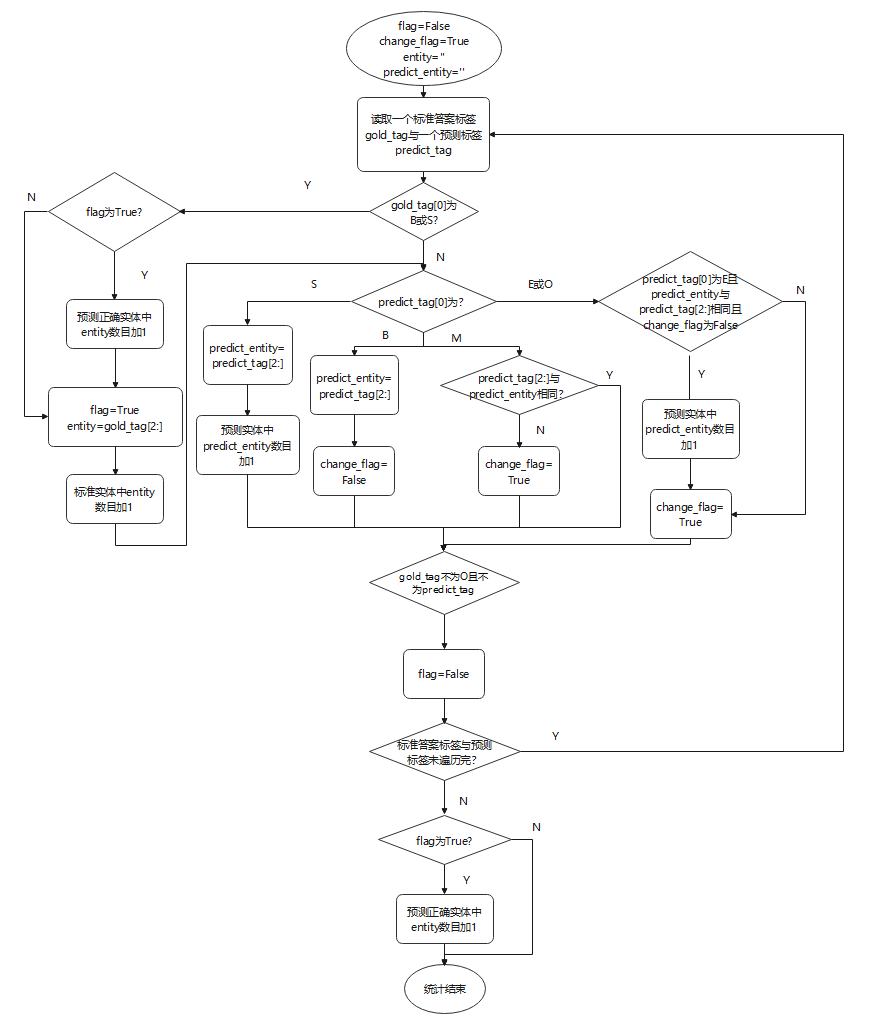
F1值=(2\*准确率P\*召回率R)/(准确率P+召回率R)

**3.2实体级命名实体识别的评价程序的实现思路**

实体级的评价程序需要分别计算各种实体以及总体的准确率P、召回率R和F1值。由上述公式知，需要分别统计预测出的实体总数、标准答案中的实体总数和预测正确的实体总数。这一部分为评价程序的核心部分，接下来将详细阐述其思路。

评价程序采用字典存储各个实体所对应的个数。同时遍历标准答案和预测答案，每当标准答案出现以B或S开头的标签，则其对应的标准答案中实体个数加一；每当预测答案出现开头为BM\*E或S的标签序列，且标签对应的实体在该序列中未发生改变时，预测出的实体个数加一。当且仅当标准答案和预测答案的标签序列完全一致，且遍历到新的一个标准答案中实体时，预测正确的实体个数加一。其中，关于标签对应实体是否变化，用标志位来记录。然后套用公式便能得到各种实体的准确率、召回率和F1值。最后和词级别的评价程序一样，采用权重平均的方式（权重为标准答案中各个实体总数）计算总体的准确率、召回率和F1值。

统计预测出的实体总数、标准答案中的实体总数和预测正确的实体总数的流程图如下所示：



统计预测出的实体总数、标准答案中的实体总数和预测正确的实体总数的伪代码如下：

①初始化：

flag = False  # 用于标志之前的实体预测是否正确

entity = ''  # 用于标志一个实际实体

predict\_entity = ''  # 用于标志预测的一个实体

change\_flag = True # 用于标志预测序列的实体类别是否变化

②遍历标准答案和预测答案实体序列：

for 标准答案和预测答案中每一个标签gold\_tag和predict\_tag：

若gold\_tag以B或S开头：

若flag为True：预测正确实体中entity数目加1

flag=True

entity=gold\_tag[2:]

标准答案实体中entity数目加1

若predict\_tag以S开头：

predict\_entity=predict\_tag[2:]

预测实体中predict\_entity数目加1

若predict\_tag以B开头：

predict\_entity=predict\_tag[2:]

change\_flag=False

若predict\_tag以M开头：

若predict\_entity与predict\_tag[2:]不同：change\_flag=True

若predict\_tag以E或O开头：

若predict\_tag以E开头且predict\_entity与predict\_tag[2:]相同且change\_flag为False：预测实体中predict\_entity数目加1

change\_flag=True

若gold\_tag不为O且与predict\_tag不同：

flag=False

③单独判断最后一个实体：

若此时flag为True: 预测正确实体中entity数目加1

**3.3评价程序调用方法**

评价程序写在类My\_metrics中，其初始化需要3个参数：golden\_tags（标准答案中所有标签结果）、predict\_tags（预测答案中所有标签结果）和remove\_O（是否移除O标签，只关心实体，默认为False，不用传入）。初始化后，调用类函数report\_scores即可得到各种实体以及总体的准确率P、召回率R和F1值。

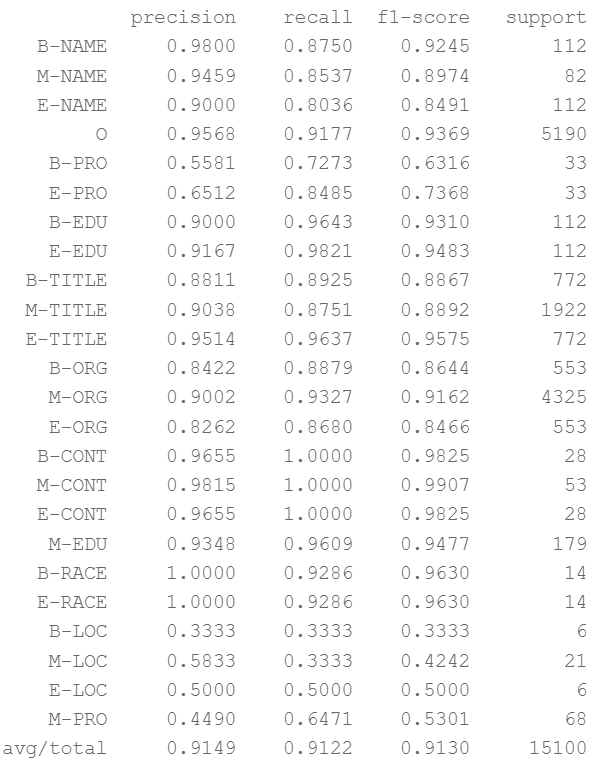
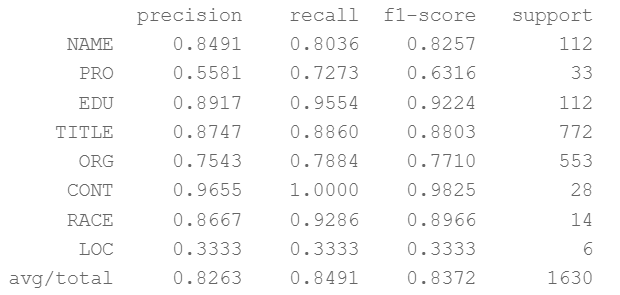
具体代码示例：

metrics = My\_metrics(test\_tag\_lists, pred\_tag\_lists, remove\_O=False)

metrics.report\_scores()

**3.4词级别和实体级别评价结果对比**

将HMM在ner\_char\_data目录下的test.txt文件上的识别结果分别按词级别和实体级别进行评价，结果如下：

可以看到，两者有较为明显的差异，具体表现为实体级别的评价指标低于词级别的评价指标。这是由于虽然大量词预测正确，但作为一个实体来考虑时，实体中只要有一个词预测错误，该实体的预测便会判为错误，而这将极大影响实体级别下的评价结果。举一个最极端的例子，若每个实体都有且仅有一个词预测错误，可以想象，词级别的评价依然能有较为理想的结果，但在实体级别下，预测正确的实体为0，所有指标将降为0，是远远低于词级别评价结果的。

# 4.基于最大熵模型的实体识别

**4.1最大熵模型原理介绍**

最大熵模型需要从训练数据中得到分类模型，该分类模型对应一个条件概率分布。

用特征函数来描述输入和输出之间的关系，定义为：

则特征函数的经验期望为：

特征函数关于分类模型的模型期望为：

若分类模型能从训练集中训练得到，则有：

若有个训练数据，则有个特征函数，且均要满足上述约束条件。

定义在条件概率分布上的条件熵为：

则最终得到的条件概率分布为：

约束条件为：

根据拉格朗日乘子法，求得最终：

**4.2基于最大熵模型的实体识别系统**

本实验中基于最大熵模型的实体识别系统包括对词特征的提取、模型的训练与测试。首先要初始化一个ME模型，其中包含一个sklearn.feature\_extraction当中的DictVectorizer实例dict\_vec（sparse参数设为True）和sklearn.linear\_model当中的LogisticRegression实例model（random\_state设为0，C设为9，solver设为liblinear）。在训练模型时，先将各个词的特征转为可计算的矩阵，再调用已初始化的ME模型进行训练。在对模型进行测试时，同样先将各个词的特征转为可计算的矩阵，再调用已训练好的ME模型进行预测。

**4.3最大熵模型利用的特征**

本实验中最大熵模型采用的特征共包含当前词及其前后各两个词的各种组合，如下所示：

'-2'+prev\_prev\_word : 1,

'-1'+prev\_word : 1,

word : 1,

'1'+next\_word : 1,

'2'+next\_next\_word : 1,

'-2-1'+prev\_prev\_word+prev\_word : 1,

prev\_word+word : 1,

word+next\_word : 1,

'12'+next\_word+next\_next\_word : 1,

prev\_prev\_word+prev\_word+word : 1,

prev\_word+word+next\_word : 1,

word+next\_word+next\_next\_word : 1,

prev\_prev\_word+prev\_word+word+next\_word : 1,

prev\_word+word+next\_word+next\_next\_word : 1,

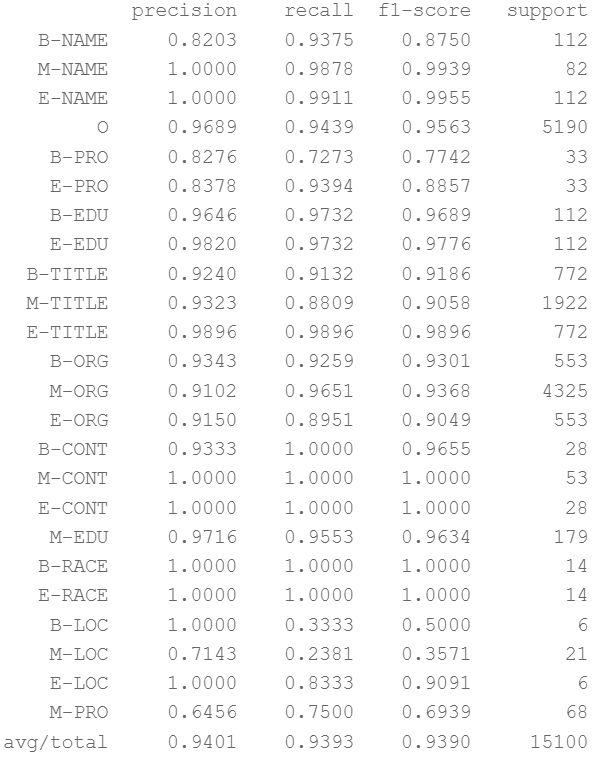
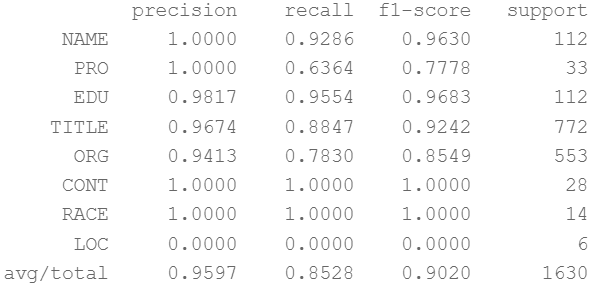
prev\_prev\_word+prev\_word+word+next\_word+next\_next\_word : 1,

'bias': 1

其中prev\_prev\_word为前前一个词、prev\_word为前一个词、word为当前词、next\_word为后一个词、next\_next\_word为后后一个词

**4.4词级别和实体级别评价结果对比**

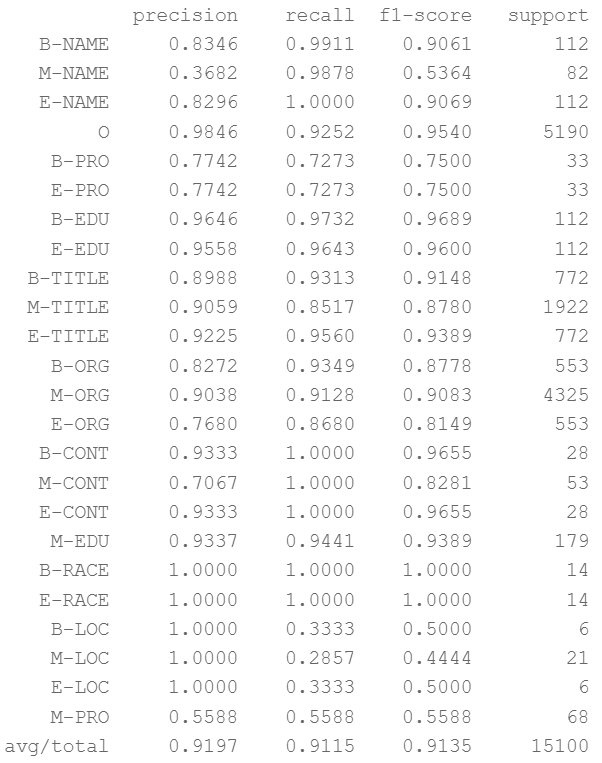
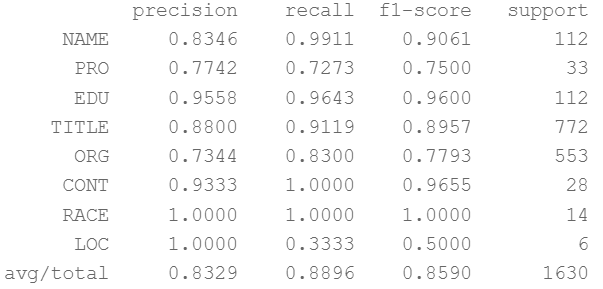
将ME模型在ner\_char\_data目录下的test.txt文件上的识别结果分别按词级别和实体级别进行评价，结果如下：

可以看到，同HMM一样，两者有较为明显的差异，具体表现为实体级别的评价指标低于词级别的评价指标。但是评价指标均优于HMM，说明ME模型在命名实体识别上比HMM更有效。

考虑到ME模型的输出存在诸如[M-ORG M-ORG E-ORG]和[B-TITLE E-NAME]这样的非法结果，可以通过加入后处理的方式有效解决这一问题。这一后处理规则由我手动编写，总体思路即遍历预测标签结果，若发现如MME和BM等非法标签序列，则将该序列前后标签改为B与E；若发现如[B-TITLE E-ORG]等标签中实体类别不一致的情况，则以第一个标签对应实体为准，将后续非法实体均改为与之相同的实体。

修改后的评价结果如下：

可以看到，和前面一样，实体级别的评价指标低于词级别的评价指标，且后处理之后的指标相较于处理前也有所下降。但是，此时的评价指标，尤其是实体级别的评价指标依然优于HMM， 这进一步证明了ME模型在命名实体识别上比HMM更有效。

# 5. HMM、ME与CRF的效果对比

**5.1ner\_clue\_data目录下数据的使用**

ner\_clue\_data目录下的数据格式相较于ner\_char\_data有一定的变化，为了能复用已实现的3种模型代码，需要在读取数据的时候进行一定的处理，使读入内存中的数据格式和ner\_char\_data一致。首先调用json来一行行地读取文件内容，使其成为易于查找的字典。然后通过字典获取每行的词数len，并将标签列表初始化为长度为len的全为O的列表。接下来再通过字典获取该行中每个实体的种类以及在句子中的index，按照index将刚刚初始化的标签列表中的相同位置改为对应的实体种类并加上BMES等前缀信息。重复上述操作直至文件内容被全部读入内存中。后面的实现就同前面一模一样了，只要调用模型、训练并测试即可。

**5.2HMM、ME和CRF的实体级别评价结果**

HMM结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| name | 0.5802 | 0.6065 | 0.5931 | 465 |
| address | 0.3668 | 0.3727 | 0.3697 | 373 |
| organization | 0.5237 | 0.5422 | 0.5328 | 367 |
| game | 0.6906 | 0.7186 | 0.7043 | 295 |
| scene | 0.4439 | 0.4354 | 0.4396 | 209 |
| book | 0.6263 | 0.4026 | 0.4901 | 154 |
| company | 0.5124 | 0.5476 | 0.5294 | 378 |
| position | 0.6413 | 0.6605 | 0.6507 | 433 |
| government | 0.5052 | 0.5951 | 0.5465 | 247 |
| movie | 0.5449 | 0.6424 | 0.5897 | 151 |
| avg/total | 0.5437 | 0.5605 | 0.5502 | 3072 |

ME结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| name | 0.8869 | 0.5226 | 0.6576 | 465 |
| address | 0.6618 | 0.2413 | 0.3536 | 373 |
| organization | 0.8588 | 0.6131 | 0.7154 | 367 |
| game | 0.8899 | 0.6847 | 0.7739 | 295 |
| scene | 0.7619 | 0.2297 | 0.3529 | 209 |
| book | 0.8696 | 0.3896 | 0.5381 | 154 |
| company | 0.8744 | 0.5159 | 0.6489 | 378 |
| position | 0.8712 | 0.6559 | 0.7484 | 433 |
| government | 0.8244 | 0.4372 | 0.5714 | 247 |
| movie | 0.9091 | 0.3974 | 0.5530 | 151 |
| avg/total | 0.8394 | 0.4932 | 0.6117 | 3072 |

ME（带后处理）结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| name | 0.6645 | 0.6688 | 0.6667 | 465 |
| address | 0.3568 | 0.4075 | 0.3805 | 373 |
| organization | 0.6632 | 0.6866 | 0.6747 | 367 |
| game | 0.6946 | 0.7864 | 0.7377 | 295 |
| scene | 0.4318 | 0.3636 | 0.3948 | 209 |
| book | 0.4509 | 0.5065 | 0.4771 | 154 |
| company | 0.5673 | 0.6243 | 0.5945 | 378 |
| position | 0.7554 | 0.7206 | 0.7376 | 433 |
| government | 0.5084 | 0.6154 | 0.5568 | 247 |
| movie | 0.4667 | 0.5099 | 0.4873 | 151 |
| avg/total | 0.5819 | 0.6113 | 0.5952 | 3072 |

CRF结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| name | 0.8082 | 0.7247 | 0.7642 | 465 |
| address | 0.5932 | 0.4692 | 0.5240 | 373 |
| organization | 0.8062 | 0.7139 | 0.7572 | 367 |
| game | 0.8161 | 0.8271 | 0.8215 | 295 |
| scene | 0.6733 | 0.4833 | 0.5627 | 209 |
| book | 0.7869 | 0.6234 | 0.6957 | 154 |
| company | 0.7955 | 0.7407 | 0.7671 | 378 |
| position | 0.8191 | 0.7113 | 0.7614 | 433 |
| government | 0.7846 | 0.7814 | 0.7830 | 247 |
| movie | 0.6818 | 0.6954 | 0.6885 | 151 |
| avg/total | 0.7642 | 0.6839 | 0.7203 | 3072 |

可以看到，经过后处理的ME模型准确率下降且召回率上升，这是由于后处理将非法标签序列修正为了合法序列，导致在现有实体级别评价体系中，预测出的实体个数和预测正确的实体数均增加。两者的共同作用使得准确率下降且召回率上升，最终的F1值也略有下降。

总的看来，CRF的结果优于ME且ME的结果优于HMM。HMM存在两个假设：一是输出观测值之间严格独立，二是状态转移过程中当前状态只与前一状态有关。实际上序列标注问题不仅和单个词相关，还和观察序列的长度，单词的上下文等等相关。ME模型解决了HMM输出独立性假设的问题，其引入自定义的特征函数能表示当前词与其前后多个词之间更复杂的关系，从而使命名实体识别效果得到提升。CRF模型不仅解决了HMM输出独立性假设的问题，还解决了ME模型的标注偏置问题，其概率模型建立在全局上，归一化时不是像ME模型一样做局部归一化，而是考虑到了数据的全局分布，于是CRF的识别效果相较于ME模型又有进一步提升。

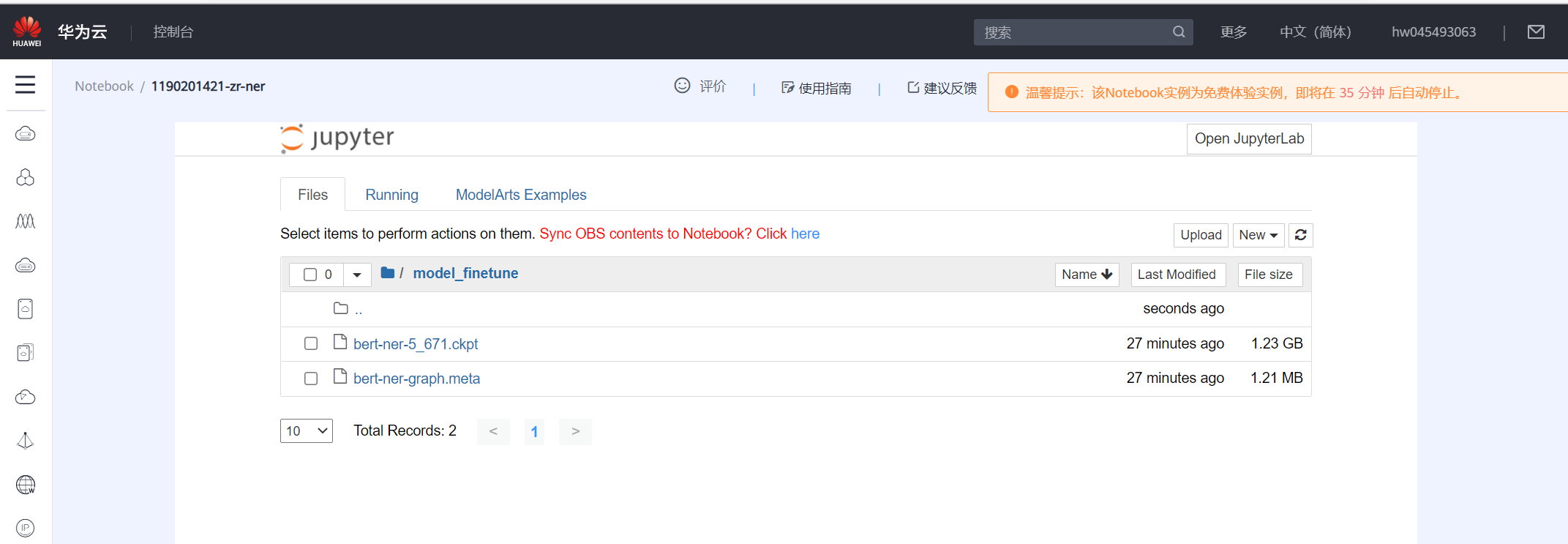
# 6. 在华为云上的计算资源进行命名实体识别

账号信息：

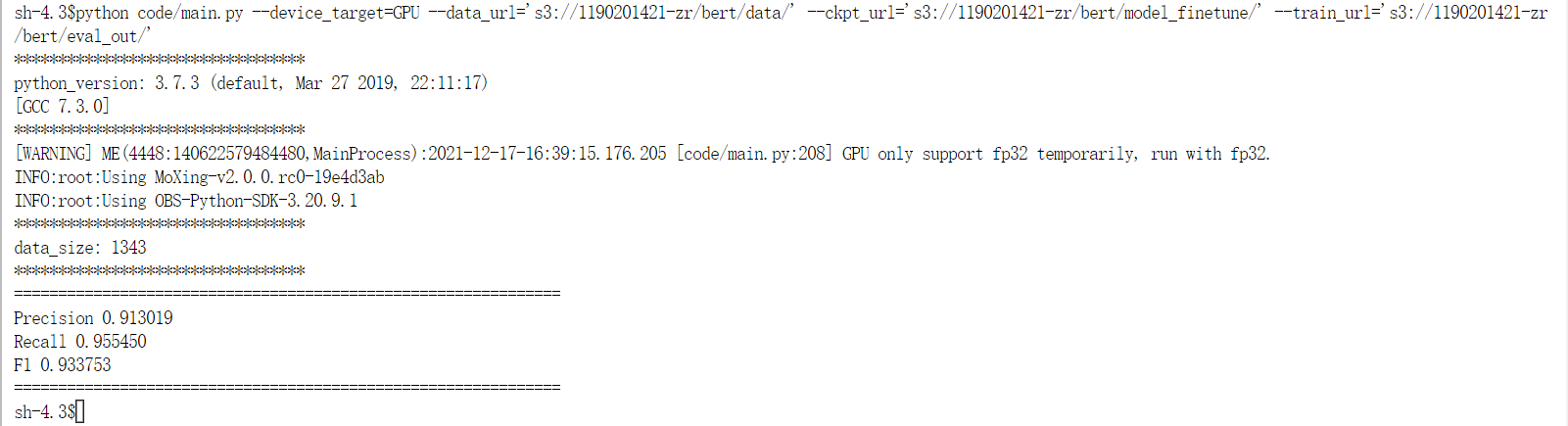


微调训练（无CRF）:

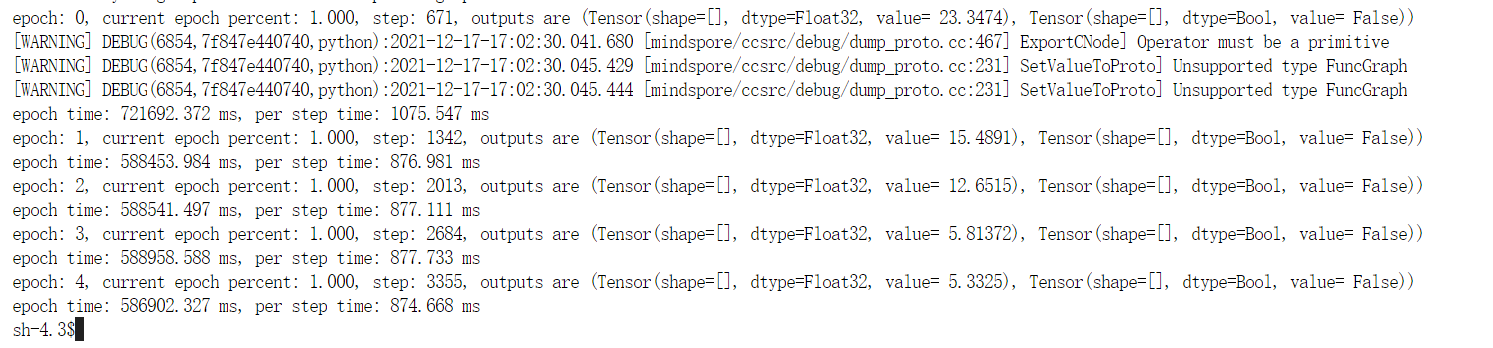


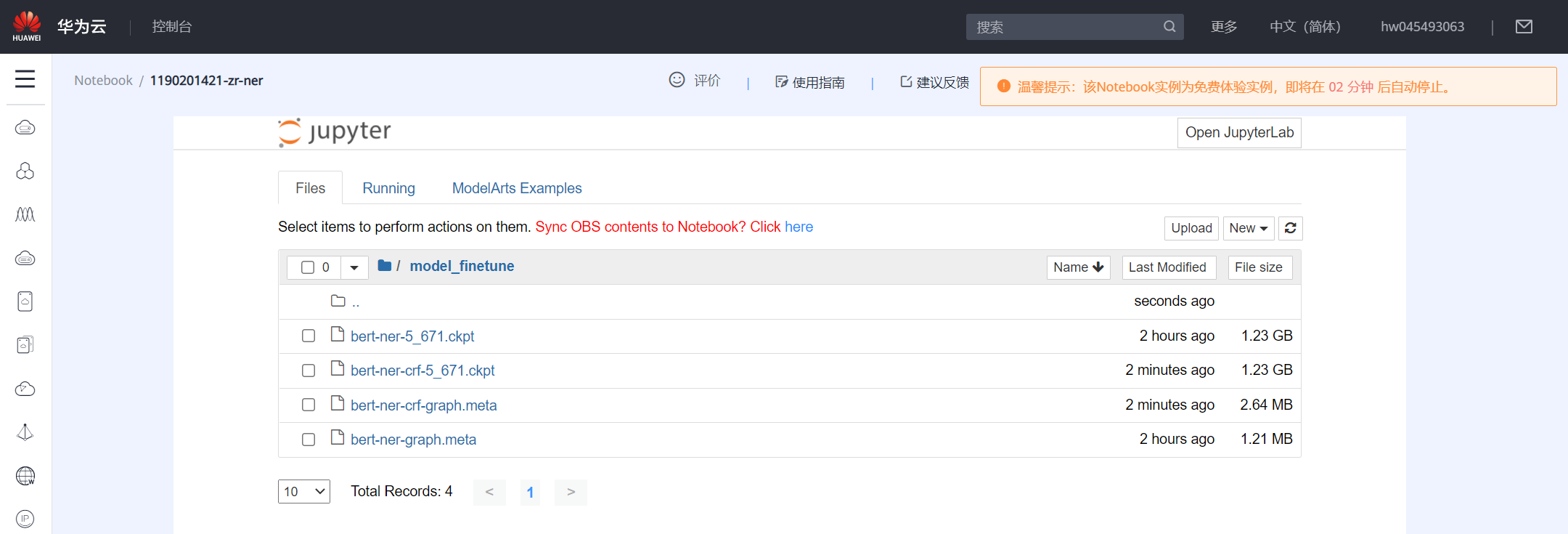


测试评价指标（无CRF）：

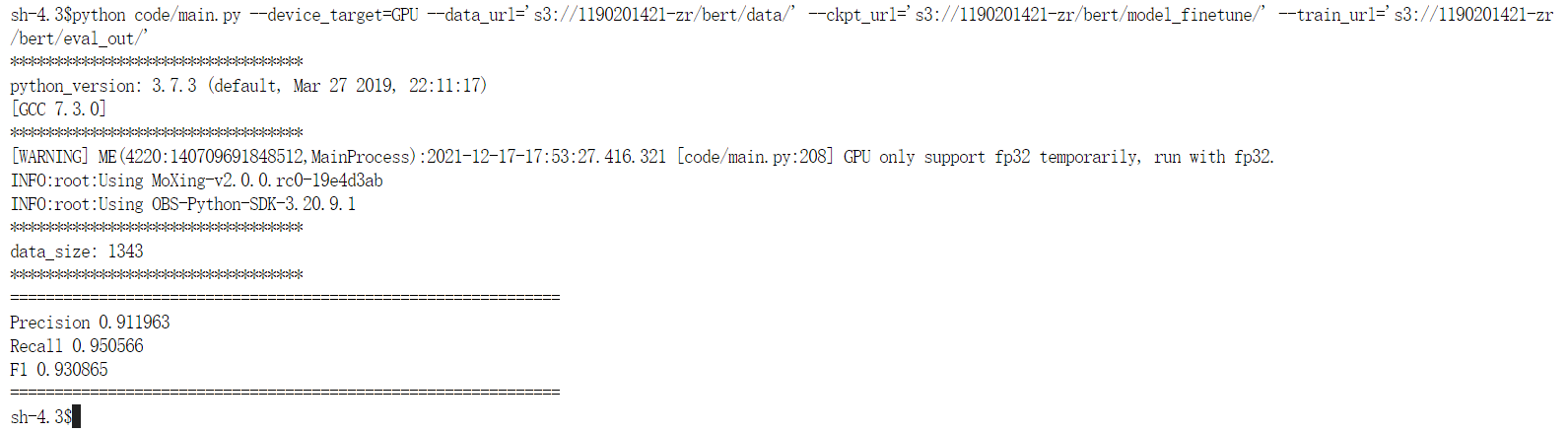


微调训练（有CRF）：

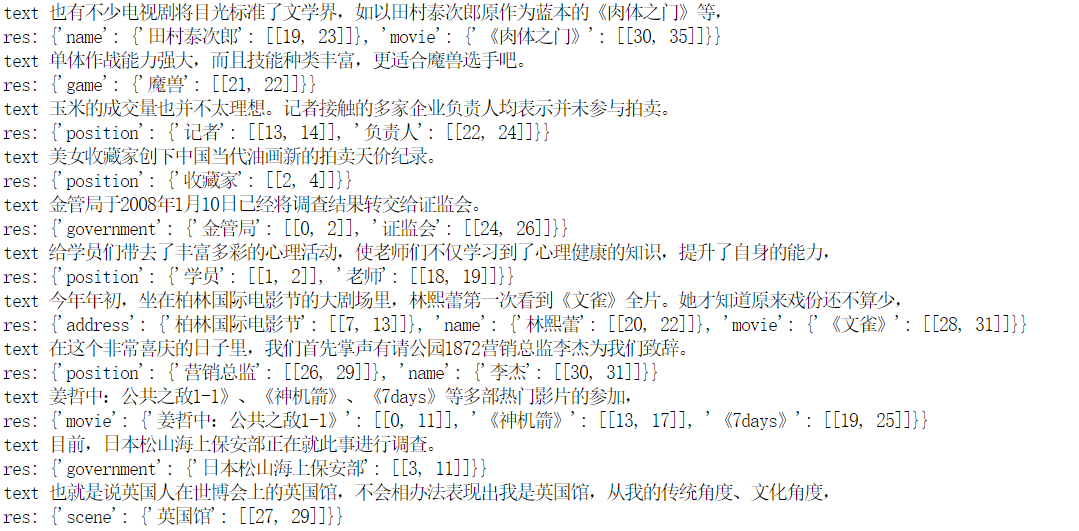


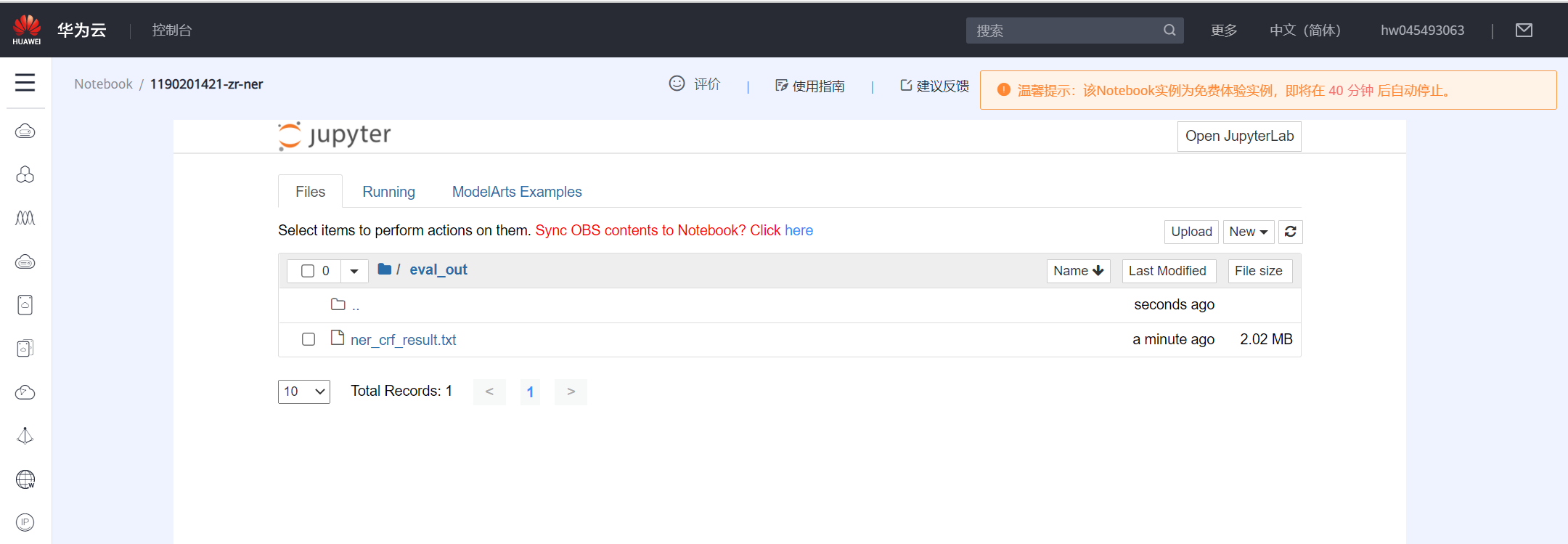


测试评价指标（有CRF）：



测试结果（有CRF）：





# 7. 实验的收获和体会

通过本次实验，我亲自动手实现了数据集的读取、数据预处理、模型训练、模型测试、后处理与评价等功能，增强了代码能力，对基于HMM、ME和CRF这三种模型的命名实体识别有了更深刻的理解，并且在对比中发现了不同实体识别方法的优缺点。同时，通过对百度AI Studio平台和华为云平台的使用，对当前国产主流人工智能平台提供的学习资源和计算资源有了一定的了解。

# 参考文献

<https://www.cnblogs.com/pinard/p/6093948.html>

<https://blog.csdn.net/liulina603/article/details/78676723>

<https://blog.csdn.net/jark_/article/details/78342644>

<https://zhuanlan.zhihu.com/p/33397147>

<https://www.cnblogs.com/hellochennan/p/6624509.html>