“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)

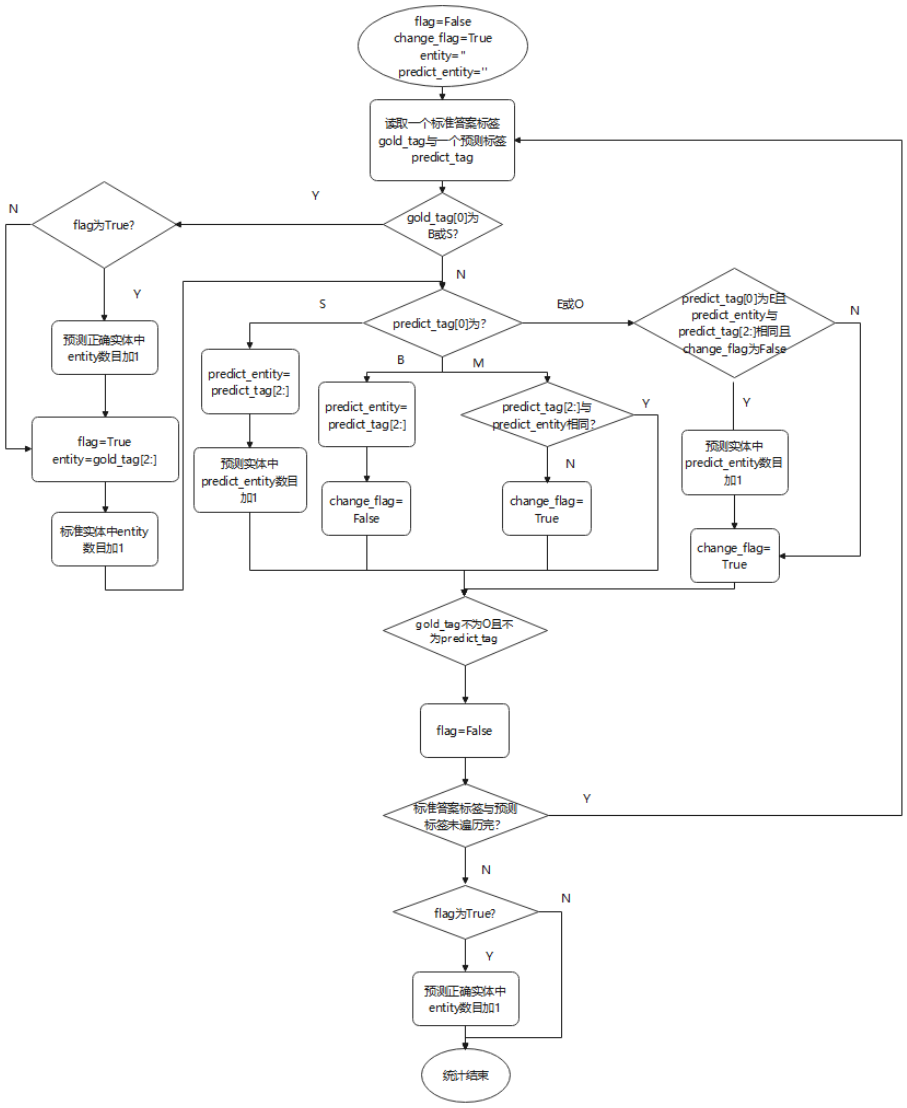
3.2 实体级命名实体识别的评价程序的实现思路

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：

	precision	recall	f1-score	support
B-NAME	0.9800	0.8750	0.9245	112
M-NAME	0.9459	0.8537	0.8974	82
E-NAME	0.9000	0.8036	0.8491	112
O	0.9568	0.9177	0.9369	5190
B-PRO	0.5581	0.7273	0.6316	33
E-PRO	0.6512	0.8485	0.7368	33
B-EDU	0.9000	0.9643	0.9310	112
E-EDU	0.9167	0.9821	0.9483	112
B-TITLE	0.8811	0.8925	0.8867	772
M-TITLE	0.9038	0.8751	0.8892	1922
E-TITLE	0.9514	0.9637	0.9575	772
B-ORG	0.8422	0.8879	0.8644	553
M-ORG	0.9002	0.9327	0.9162	4325
E-ORG	0.8262	0.8680	0.8466	553
B-CONT	0.9655	1.0000	0.9825	28
M-CONT	0.9815	1.0000	0.9907	53
E-CONT	0.9655	1.0000	0.9825	28
M-EDU	0.9348	0.9609	0.9477	179
B-RACE	1.0000	0.9286	0.9630	14
E-RACE	1.0000	0.9286	0.9630	14
B-LOC	0.3333	0.3333	0.3333	6
M-LOC	0.5833	0.3333	0.4242	21
E-LOC	0.5000	0.5000	0.5000	6
M-PRO	0.4490	0.6471	0.5301	68
avg/total	0.9149	0.9122	0.9130	15100

	precision	recall	f1-score	support
NAME	0.8491	0.8036	0.8257	112
PRO	0.5581	0.7273	0.6316	33
EDU	0.8917	0.9554	0.9224	112
TITLE	0.8747	0.8860	0.8803	772
ORG	0.7543	0.7884	0.7710	553
CONT	0.9655	1.0000	0.9825	28
RACE	0.8667	0.9286	0.8966	14
LOC	0.3333	0.3333	0.3333	6
avg/total	0.8263	0.8491	0.8372	1630

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$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$中得到分类模型，该分类模型对应一个条件概率分布$p(y|x)$。

用特征函数$f(x, y)$来描述输入x和输出y之间的关系，定义为：

$$f(x,y) = \begin{cases} 1 & x\text{与}y\text{满足某个关系} \\ 0 & \text{否则} \end{cases}$$

则特征函数的经验期望为：

$$\tilde{p}(f) = \sum_{x,y} \tilde{p}(x,y) f(x,y)$$

特征函数关于分类模型的模型期望为：

$$p(f) = \sum_{x,y} \tilde{p}(x) p(y|x) f(x,y)$$

若分类模型能从训练集中训练得到，则有：

$$\tilde{p}(f) = p(f)$$

若有n个训练数据，则有n个特征函数$f_i(i = 1,2, \dots, n)$，且均要满足上述约束条件。

定义在条件概率分布$p(y|x)$上的条件熵为：

$$H\big(p(y|x)\big) = -\sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x)$$

则最终得到的条件概率分布为：

$$p^* = \underset{p(y|x)}{\operatorname{argmax}} H(p(y|x))$$

约束条件为：

$$p(y|x) \geq 0$$

$$\sum_y p(y|x) = 1$$

$$\tilde{p}(f) = p(f)$$

根据拉格朗日乘子法，求得最终：

$$p(y|x) = \frac{1}{Z(x)} exp \sum_{i=1}^{n} \lambda_i f_i(x,y)$$

$$Z(x) = \sum_{y=1}^{n} exp \sum_{i=1}^{n} \lambda_i f_i(x,y)$$

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：

	precision	recall	f1-score	support
B-NAME	0.8203	0.9375	0.8750	112
M-NAME	1.0000	0.9878	0.9939	82
E-NAME	1.0000	0.9911	0.9955	112
O	0.9689	0.9439	0.9563	5190
B-PRO	0.8276	0.7273	0.7742	33
E-PRO	0.8378	0.9394	0.8857	33
B-EDU	0.9646	0.9732	0.9689	112
E-EDU	0.9820	0.9732	0.9776	112
B-TITLE	0.9240	0.9132	0.9186	772
M-TITLE	0.9323	0.8809	0.9058	1922
E-TITLE	0.9896	0.9896	0.9896	772
B-ORG	0.9343	0.9259	0.9301	553
M-ORG	0.9102	0.9651	0.9368	4325
E-ORG	0.9150	0.8951	0.9049	553
B-CONT	0.9333	1.0000	0.9655	28
M-CONT	1.0000	1.0000	1.0000	53
E-CONT	1.0000	1.0000	1.0000	28
M-EDU	0.9716	0.9553	0.9634	179
B-RACE	1.0000	1.0000	1.0000	14
E-RACE	1.0000	1.0000	1.0000	14
B-LOC	1.0000	0.3333	0.5000	6
M-LOC	0.7143	0.2381	0.3571	21
E-LOC	1.0000	0.8333	0.9091	6
M-PRO	0.6456	0.7500	0.6939	68
avg/total	0.9401	0.9393	0.9390	15100

	precision	recall	f1-score	support
NAME	1.0000	0.9286	0.9630	112
PRO	1.0000	0.6364	0.7778	33
EDU	0.9817	0.9554	0.9683	112
TITLE	0.9674	0.8847	0.9242	772
ORG	0.9413	0.7830	0.8549	553
CONT	1.0000	1.0000	1.0000	28
RACE	1.0000	1.0000	1.0000	14
LOC	0.0000	0.0000	0.0000	6
avg/total	0.9597	0.8528	0.9020	1630

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：

	precision	recall	f1-score	support
B-NAME	0.8346	0.9911	0.9061	112
M-NAME	0.3682	0.9878	0.5364	82
E-NAME	0.8296	1.0000	0.9069	112
O	0.9846	0.9252	0.9540	5190
B-PRO	0.7742	0.7273	0.7500	33
E-PRO	0.7742	0.7273	0.7500	33
B-EDU	0.9646	0.9732	0.9689	112
E-EDU	0.9558	0.9643	0.9600	112
B-TITLE	0.8988	0.9313	0.9148	772
M-TITLE	0.9059	0.8517	0.8780	1922
E-TITLE	0.9225	0.9560	0.9389	772
B-ORG	0.8272	0.9349	0.8778	553
M-ORG	0.9038	0.9128	0.9083	4325
E-ORG	0.7680	0.8680	0.8149	553
B-CONT	0.9333	1.0000	0.9655	28
M-CONT	0.7067	1.0000	0.8281	53
E-CONT	0.9333	1.0000	0.9655	28
M-EDU	0.9337	0.9441	0.9389	179
B-RACE	1.0000	1.0000	1.0000	14
E-RACE	1.0000	1.0000	1.0000	14
B-LOC	1.0000	0.3333	0.5000	6
M-LOC	1.0000	0.2857	0.4444	21
E-LOC	1.0000	0.3333	0.5000	6
M-PRO	0.5588	0.5588	0.5588	68
avg/total	0.9197	0.9115	0.9135	15100

	precision	recall	f1-score	support
NAME	0.8346	0.9911	0.9061	112
PRO	0.7742	0.7273	0.7500	33
EDU	0.9558	0.9643	0.9600	112
TITLE	0.8800	0.9119	0.8957	772
ORG	0.7344	0.8300	0.7793	553
CONT	0.9333	1.0000	0.9655	28
RACE	1.0000	1.0000	1.0000	14
LOC	1.0000	0.3333	0.5000	6
avg/total	0.8329	0.8896	0.8590	1630

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。

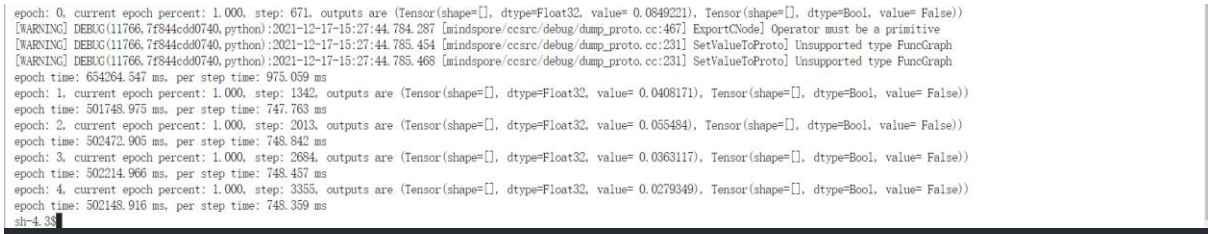
总的看来，CRF 的结果优于 ME 且 ME 的结果优于 HMM。HMM 存在两个假设：一是输出观测值之间严格独立，二是状态转移过程中当前状态只与前一状态有关。实际上序列标注问题不仅和单个词相关，还和观察序列的长度，单词的上下文等等相关。ME 模型解决了 HMM 输出独立性假设的问题，其引入自定义的特征函数能表示当前词与其前后多个词之间更复杂的关系，从而使命名实体识别效果得到提升。CRF 模型

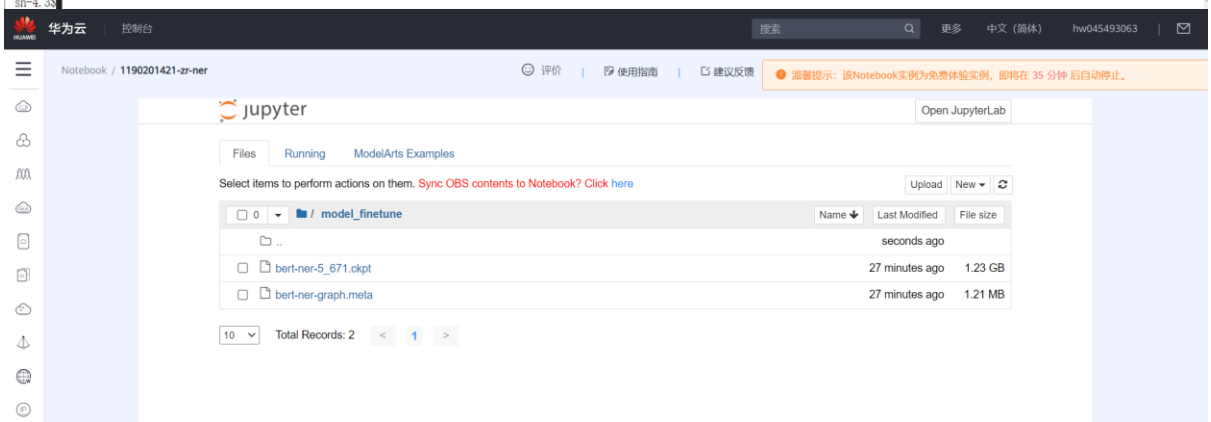
不仅解决了 HMM 输出独立性假设的问题，还解决了 ME 模型的标注偏置问题，其概率模型建立在全局上，归一化时不是像 ME 模型一样做局部归一化，而是考虑到了数据的全局分布，于是 CRF 的识别效果相较于 ME 模型又有进一步提升。

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）：



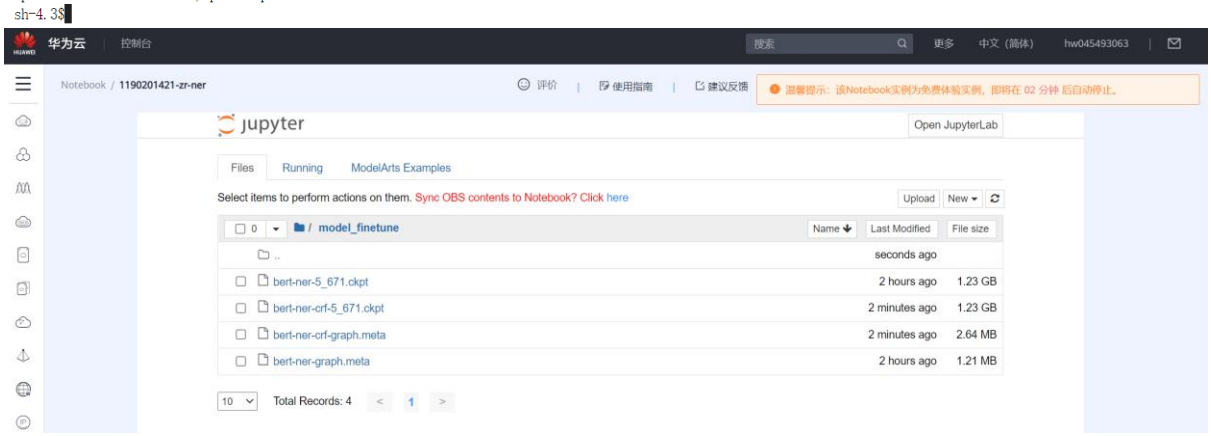


测试评价指标（无 CRF）：

```
sh-4.3$python code/main.py --device_target=GPU --data_url=' s3://1190201421-zr/bert/data/' --ckpt_url=' s3://1190201421-zr/bert/model_finetune/' --train_url=' s3://1190201421-zr
/bert/eval_out/'
==================================
python_version: 3.7.3 (default, Mar 27 2019, 22:11:17)
[GCC 7.3.0]
==================================
[WARNING] ME(4448:140622579484480,MainProcess):2021-12-17-16:39:15.176.205 [code/main.py:208] GPU only support fp32 temporarily, run with fp32.
INFO:root:Using MoXing-v2.0.0.rc0-19e4d3ab
INFO:root:Using OBS-Python-SDK-3.20.9.1
==================================
data_size: 1343
==================================
============================================================
Precision 0.913019
Recall 0.955450
F1 0.933753
============================================================
sh-4.3$
```

微调训练（有 CRF）：

```
epoch: 0, current epoch percent: 1.000, step: 671, outputs are (Tensor(shape=[], dtype=Float32, value= 23.3474), Tensor(shape=[], dtype=Bool, value= False))
[WARNING] DEBUG(6854,7f847e440740,python):2021-12-17-17:02:30.041.680 [mindspore/ccsrc/debug/dump_proto.cc:467] ExportCNode] Operator must be a primitive
[WARNING] DEBUG(6854,7f847e440740,python):2021-12-17-17:02:30.045.429 [mindspore/ccsrc/debug/dump_proto.cc:231] SetValueToProto] Unsupported type FuncGraph
[WARNING] DEBUG(6854,7f847e440740,python):2021-12-17-17:02:30.045.444 [mindspore/ccsrc/debug/dump_proto.cc:231] SetValueToProto] Unsupported type FuncGraph
epoch time: 721692.372 ms, per step time: 1075.547 ms
epoch: 1, current epoch percent: 1.000, step: 1342, outputs are (Tensor(shape=[], dtype=Float32, value= 15.4891), Tensor(shape=[], dtype=Bool, value= False))
epoch time: 588453.984 ms, per step time: 876.981 ms
epoch: 2, current epoch percent: 1.000, step: 2013, outputs are (Tensor(shape=[], dtype=Float32, value= 12.6515), Tensor(shape=[], dtype=Bool, value= False))
epoch time: 588541.497 ms, per step time: 877.111 ms
epoch: 3, current epoch percent: 1.000, step: 2684, outputs are (Tensor(shape=[], dtype=Float32, value= 5.81372), Tensor(shape=[], dtype=Bool, value= False))
epoch time: 588958.588 ms, per step time: 877.733 ms
epoch: 4, current epoch percent: 1.000, step: 3355, outputs are (Tensor(shape=[], dtype=Float32, value= 5.3325), Tensor(shape=[], dtype=Bool, value= False))
epoch time: 586902.327 ms, per step time: 874.668 ms
sh-4.3$
```

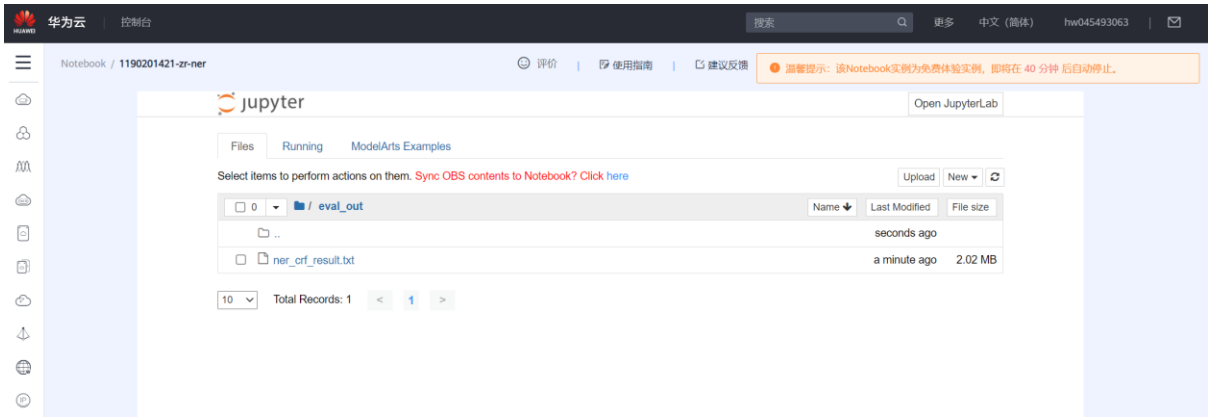


测试评价指标（有 CRF）：

```
sh-4.3$python code/main.py --device_target=GPU --data_url=' s3://1190201421-zr/bert/data/' --ckpt_url=' s3://1190201421-zr/bert/model_finetune/' --train_url=' s3://1190201421-zr
/bert/eval_out/'
==================================
python_version: 3.7.3 (default, Mar 27 2019, 22:11:17)
[GCC 7.3.0]
==================================
[WARNING] ME(4220:140709691848512,MainProcess):2021-12-17-17:53:27.416.321 [code/main.py:208] GPU only support fp32 temporarily, run with fp32.
INFO:root:Using MoXing-v2.0.0.rc0-19e4d3ab
INFO:root:Using OBS-Python-SDK-3.20.9.1
==================================
data_size: 1343
==================================
============================================================
Precision 0.911963
Recall 0.950566
F1 0.930865
============================================================
sh-4.3$
```

测试结果（有 CRF）：

```
text 也有不少电视剧将目光标准了文学界，如以田村泰次郎原作为蓝本的《肉体之门》等，
res: {'name': {'田村泰次郎': [[19, 23]]}, 'movie': {'《肉体之门》': [[30, 35]]}}
text 单体作战能力强大，而且技能种类丰富，更适合魔兽选手吧。
res: {'game': {'魔兽': [[21, 22]]}}
text 玉米的成交量也并不太理想。记者接触的多家企业负责人均表示并未参与拍卖。
res: {'position': {'记者': [[13, 14]], '负责人': [[22, 24]]}}
text 美女收藏家创下中国当代油画新的拍卖天价纪录。
res: {'position': {'收藏家': [[2, 4]]}}
text 金管局于2008年1月10日已经将调查结果转交给证监会。
res: {'government': {'金管局': [[0, 2]], '证监会': [[24, 26]]}}
text 给学员们带去了丰富多彩的心理活动，使老师们不仅学习到了心理健康的知识，提升了自身的能力，
res: {'position': {'学员': [[1, 2]], '老师': [[18, 19]]}}
text 今年年初，坐在柏林国际电影节的大剧场里，林熙蕾第一次看到《文雀》全片。她才知道原来戏份还不算少，
res: {'address': {'柏林国际电影节': [[7, 13]]}, 'name': {'林熙蕾': [[20, 22]]}, 'movie': {'《文雀》': [[28, 31]]}}
text 在这个非常喜庆的日子里，我们首先掌声有请公园1872营销总监李杰为我们致辞。
res: {'position': {'营销总监': [[26, 29]]}, 'name': {'李杰': [[30, 31]]}}
text 姜哲中：公共之敌1-1》、《神机箭》、《7days》等多部热门影片的参加，
res: {'movie': {'姜哲中：公共之敌1-1》': [[0, 11]], '《神机箭》': [[13, 17]], '《7days》': [[19, 25]]}}
text 目前，日本松山海上保安部正在就此事进行调查。
res: {'government': {'日本松山海上保安部': [[3, 11]]}}
text 也就是说英国人在世博会上的英国馆，不会相办法表现出我是英国馆，从我的传统角度、文化角度，
res: {'scene': {'英国馆': [[27, 29]]}}
```



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