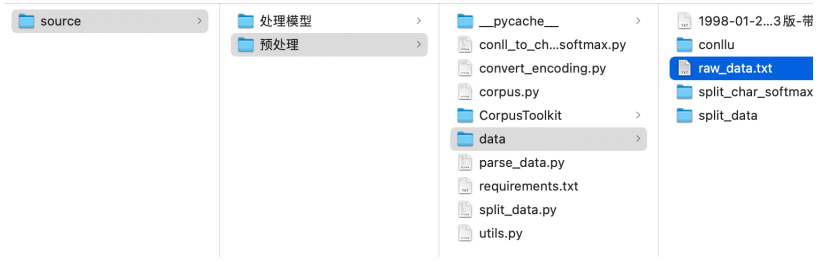
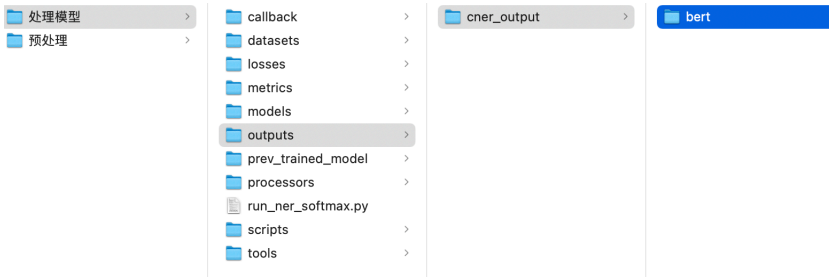
知识工程第一次大作业报告

姓名：赵炫皓 学号:1120200603

项目源码：





1 NER简介

NER（命名实体识别）又称作专名识别，是自然语言处理中常见且基本的一项任务。命名实体识别通常指的是文本中具有特殊意义或指代性强的实体，通常包括人名，地名，机构名，时间，专有名词等。NER系统就是从非结构化的文本中抽取出上述实体，并且可以按照业务需求识别出更多类型的实体。

2 NER问题的分解

NER问题目标是从文本中抽取特定需求实体的文本片段。针对这个任务，通常使用基于规则的方法和基于模型的方法。由于基于规则的方法适用性差，维护成本高，因此一般只用于处理简单的NER问题。

2.2 基于模型的方法

从模型的角度来看，NER问题实际上是序列标注问题，序列标注问题指的是模型的输入是一个序列，包括文字，时间等，输出也是一个序列。针对输入序列的每一个单元，输出一个特定的标签。以中文分词任务举例，例如输入序列的每一个单元，输出一个特定的标签。源码中使用的是BIO标注方法，即B表示这个字是词的开始Begin，I表示这个字是词的中间部分。

2.2.1 softmax函数

softmax用于多分类过程中，它将多个神经元输出，映射到(0 , 1)区间内，可以看成概率来理解，从而进行多分类。

假设我们有一个数组，V，Vi表示V中的第i个元素，那么这个元素的softmax值就是:

$$S_i = \frac{e^i}{\sum_j e^j}$$

softmax将原来的值通过softmax函数作用映射为 （0 ， 1）的值，而这些值的累和是1（满足概率的性质),那么可以将其理解为概率，在最后输出结点时选取概率最大的结点作为我们预测的目标。

当我们对分类的Loss进行改进的时候，我们需要通过梯度下降，每次优化一个step大小的梯度，这个时候要求Loss对每个权重矩阵求偏导，用链式法则。使用softmax函数后，梯度求导过程非常方便。

3基于BERT + softmax模型打造中文NER系统

3.1 明确标注目标

NER可以根据业务需要标注各种不同类型的实体，因此首先要明确需要抽取的实体类型。一般通用场景下，最常提取的是时间，任务，地点以及机构名。作业任务中提取了PERSON,LOCATION,ORGNAZATION三种实体。

3.2 数据以及工具准备

明确任务后需要训练数据和模型工具。首先下载人民日报1998中文标注语料库，进行处理的环境配置如下:

```
1.1.0 =< PyTorch < 1.5.0
cuda=9.0
python3.6+
Click==7.0
joblib==0.13.2
nltk==3.4.5
numpy==1.17.2
pandas==0.25.1
ply==3.11
python-dateutil==2.8.0
pytz==2019.2
scikit-learn==0.21.3
scipy==1.3.1
six==1.12.0
tokenizer-tools==0.15.3
```

3.3 数据的预处理

人民日报1998语料库下载完毕后，解压打开“199801.txt”这个文件，首先用预处理中python文件convert_encoding.py将编码格式转换成UTF-8格式在data文件夹中得到raw_data.txt，方便使用文字编辑器读取。我们需要的提取的实体是人名，地名，组织机构名，根据1998语料库的词性标记说明，对应的词性依次为nr、ns、nt。运行python文件parse_data.py对语料库进行处理。处理的要点有四则：

1. 1998语料库标注人名时，将姓和名分开标注，因此需要合并姓名，比如 `兰/nrf 红光/nrg` 应该合并成 `兰红光/nr`
2. 中括号括起来的几个词表示大粒度分词，表意能力更强，需要将括号内内容合并，比如 `[中央/n 人民/n 广播/vn 电台/n]nt` 应该合并成 `中央人民广播电台/nt`
3. 时间合并,例如将“1997年/t 3月/t” 合并成”1997年3月/t”
4. 全角字符统一转为半角字符，尤其是数字的表示，比如：`１９９８年/t` 应该转换成 `1998年/t`

运行parse.py文件，raw_data.txt转换为conllu格式的文件data_False-True-True-True-True-True-False.conllu

通过语料预处理得到结果如图1，2所示。

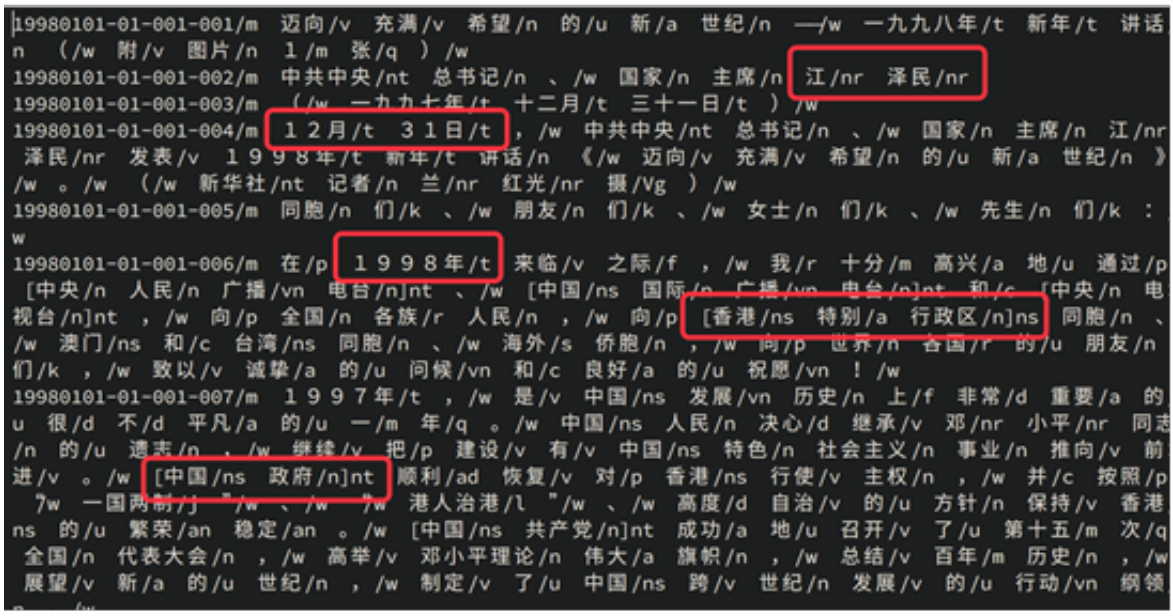


图1.预处理前

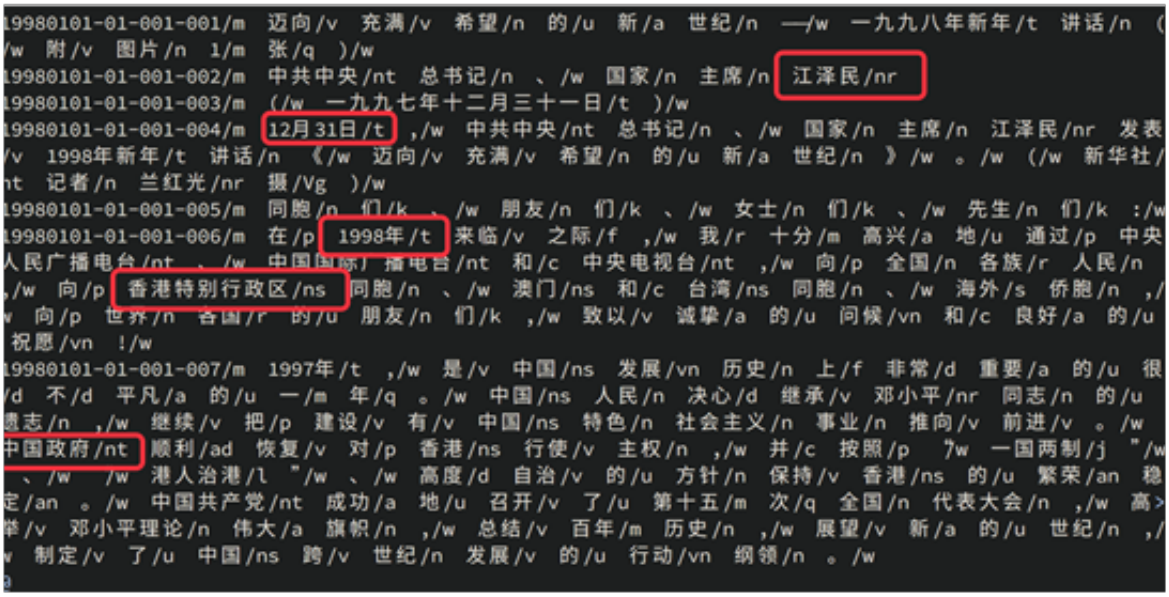


图2.预处理后

3.4 模型训练

根据NER任务需求以及BERT+softmax训练要求，训练模型需要3个步骤：1，确定标签体系 2，处理训练数据文件 3，训练模型

3.4.1 确定标签体系

大部分情况下，标签体系越复杂准确度也越高，但相应的训练时间也会增加。本次处理使用BIO标签体系。

Tokens	IO	BIO	BMEWO	BMEWO+
昨	O	O	O	O
天	O	O	O	O
，	O	O	O	O_PERSON
李	I_PERSON	B_PERSON	B_PERSON	B_PERSON
晓	I_PERSON	I_PERSON	M_PERSON	M_PERSON
明	I_PERSON	I_PERSON	E_PERSON	E_PERSON
前	O	O	O	PERSON_O
往	O	O	O	O_LOCATION
上	I_LOCATION	B_LOCATION	B_LOCATION	B_LOCATION
海	I_LOCATION	I_LOCATION	E_LOCATION	E_LOCATION
。	O	O	O	LOCATION_O

图3.标签体系

3.4.2 数据标注处理

得到data_False-True-True-True-True-True-False.conllu文件后，用split_data.py把文件分为训练集，验证集，和测试集,分别为split_data文件夹的dev.conllu,test.conllu,train.conllu。然后使用conll_to_char_softmax.py文件将conllu格式文件用tokenizer_tools库所自带的BILUOEncoderDecoder将数据转化为BILUO标注再转化为BIO标注。得到如图所示的标注文件。

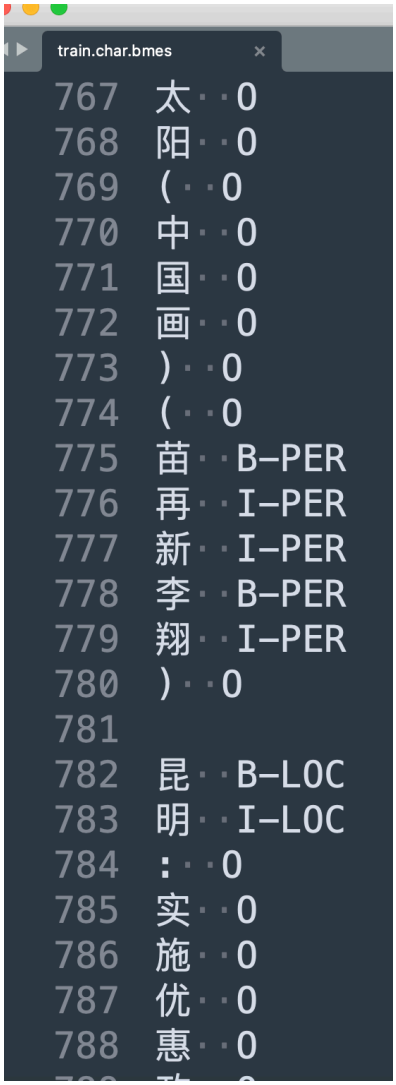


图4.BIO标注

自此，对数据的预处理全部结束，下面进行对数据的训练。

3.4.3 训练数据生成

Run_ner_softmax.py包含了处理BERT+softmax的核心代码。

其中prev_trained_model的文件格式为

```
├── prev_trained_model
|   └── bert_base
|   |   └── pytorch_model.bin
|   |   └── config.json
|   |   └── vocab.txt
|   |   └── ......
```

处理过程如下：

```
guid: train-3
tokens: [CLS] 与 此 同 时 ， 金 融 风 暴 又 席 卷 韩 国 ， 经 济 形 势 随 之 急 剧 恶 化 。 灾 难 临 头 ，
一 年 前 刚 刚 加 入 了 欧 洲 经 合 组 织 这 个 富 国 俱 乐 部 的 韩 国 ， 不 得 不 忍 痛 向 国 际 货 币 基
金 组 织 求 援 。 经 过 一 番 紧 张 谈 判 ， 国 际 货 币 基 金 组 织 同 意 向 韩 国 提 供 多 达 5 7 0 亿
美 元 的 援 助 。 如 有 必 要 ， 美 国 还 答 应 同 其 它 国 家 一 起 [SEP]
input_ids: 101 680 3634 1398 3198 117 7032 6084 7599 3274 1348 2375 1318 7506 1744 117 5307
3845 2501 1232 7390 722 2593 1196 2626 1265 511 4135 7410 707 1928 117 671 2399 1184 1157 115
1217 1057 749 3616 3828 5307 1394 5299 5302 6821 702 2168 1744 936 727 6956 4638 7506 1744 11
679 2533 679 2556 4578 1403 1744 7354 6573 2355 1825 7032 5299 5302 3724 3001 511 5307 6814
671 4528 5165 2476 6448 1161 117 1744 7354 6573 2355 1825 7032 5299 5302 1398 2692 1403 7506
1744 2990 897 1914 6809 126 128 121 783 5401 1039 4638 3001 1221 511 1963 3300 2553 6206 117
5401 1744 6820 5031 2418 1398 1071 2124 1744 2157 671 6629 102
input_mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
segment_ids: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
label_ids: 17 17 17 17 17 17 17 17 17 17 17 17 17 3 11 17 17 17 17 17 17 17 17 17 17 17 17 17
17 17 17 17 17 17 17 17 17 17 17 3 11 17 17 17 17 17 17 17 17 17 17 17 17 3 11 17 17 17 17
17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
17 17 17 3 11 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 3 11 17 17 17 17 17 17
17 17 17 17 17
*** Example ***
guid: train-4
tokens: [CLS] 5 、 我 们 的 队 伍 向 太 阳 （ 中 国 画 ）（ 苗 再 新 李 翔 ）[SEP]
input_ids: 101 126 510 2769 812 4638 7339 824 1403 1922 7345 113 704 1744 4514 114 113 5728
1086 3173 3330 5425 114 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

图5.BERT + Softmax过程

对模型进行训练后，得到的BERT + Softmax模型的训练效率结果如下：

	Accuracy (entity)	Recall (entity)	F1 score (entity)
BERT+Softmax	0.7897	0.8031	0.7963

表1.BERT + Softmax

重复步骤3次得到的训练效率如下：

```
03/28/2022 00:01:33 - INFO - root -   Evaluate the following checkpoints: ['/Use
rs/myfile/Desktop/BERT-NER-Pytorch-master/outputs/cner_output/bert']
03/28/2022 00:01:34 - INFO - root -   Loading features from cached file /Users/m
yfile/Desktop/BERT-NER-Pytorch-master/datasets/cner/cached_soft-dev_bert-base-ch
inese_512_cner
03/28/2022 00:01:35 - INFO - root -   ***** Running evaluation  *****
03/28/2022 00:01:35 - INFO - root -     Num examples = 171
03/28/2022 00:01:35 - INFO - root -     Batch size = 24
[Evaluating] 8/8 [==============================] 12.5s/step2
03/28/2022 00:03:15 - INFO - root -

03/28/2022 00:03:15 - INFO - root -   ***** Eval results  *****
03/28/2022 00:03:15 - INFO - root -    acc: 0.9385 - recall: 0.9639 - f1: 0.9510
 - loss: 0.0227
03/28/2022 00:03:15 - INFO - root -   ***** Entity results  *****
03/28/2022 00:03:15 - INFO - root -   ******* LOC results ********
03/28/2022 00:03:15 - INFO - root -    acc: 0.9141 - recall: 0.9744 - f1: 0.9433

03/28/2022 00:03:15 - INFO - root -   ******* ORG results ********
03/28/2022 00:03:15 - INFO - root -    acc: 0.9524 - recall: 0.8333 - f1: 0.8889

03/28/2022 00:03:15 - INFO - root -   ******* PER results ********
03/28/2022 00:03:15 - INFO - root -    acc: 0.9860 - recall: 0.9658 - f1: 0.9758
```

图6.BERT + Softmax训练结果

4. 总结

本次任务基于BERT + Softmax模型，完成了对于人民日报1998版语料的NER。在本次任务中，首先对python语言进行复习，并熟悉pytorch和pandas等库，在此基础上理解了NLP任务中比较基本的NER任务的处理思路与处理方法。