**细粒度命名实体识别**

## 任务简介

简单来说，命名实体识别是一个序列标注任务，给定一句话，需要标注其中每个实体的类型。比如说输入是：

小明 在 北京大学 的 燕园 看了

那输出应该是：

PER – ORG – LOC –

当然这是一个词级别的示意，实际我们都是做到token级别的。 模型的性能将通过与ground truth比较计算F1值来进行测定。

在本次任务中，我们将利用BERT等预训练模型来完成这个任务。不过鉴于我们已经学习过bert和一定的数据预处理，本次作业要求完成一定的算法。

## 任务指标

1. 补全models/layers/crf.py中的\_viterbi\_decode函数（维比特解码）
2. 还是同一个文件，仿照上个函数完成\_viterbi\_decode\_nbest函数的编写
3. 补全losses/label\_smoothing.py中的Label Smoothing Loss
4. 补全losses/focal\_loss.py中的FocalLoss
5. 按照说明运行实验，看看效果如何，下面给出了batch\_size=24时一个测试的效果。在给的脚本中已经把batch size设置成了8，linux上测试占用约3.8G显存，大家如果自己电脑显卡好一些应该可以亲自动手跑一跑，并看到结果。如果没有合适的条件，可以尝试用CPU运行，能跑通也可以。祝各位顺利。

\*\*\*\*\* Eval results \*\*\*\*\*

acc: 0.7848 - recall: 0.7992 - f1: 0.7919 - loss: 5.9302

\*\*\*\*\* Entity results \*\*\*\*\*

\*\*\*\*\*\*\* address results \*\*\*\*\*\*\*\*

acc: 0.6740 - recall: 0.5764 - f1: 0.6214

\*\*\*\*\*\*\* book results \*\*\*\*\*\*\*\*

acc: 0.8182 - recall: 0.8182 - f1: 0.8182

\*\*\*\*\*\*\* company results \*\*\*\*\*\*\*\*

acc: 0.7863 - recall: 0.7884 - f1: 0.7873

\*\*\*\*\*\*\* game results \*\*\*\*\*\*\*\*

acc: 0.7522 - recall: 0.8847 - f1: 0.8131

\*\*\*\*\*\*\* government results \*\*\*\*\*\*\*\*

acc: 0.7801 - recall: 0.8907 - f1: 0.8318

\*\*\*\*\*\*\* movie results \*\*\*\*\*\*\*\*

acc: 0.8740 - recall: 0.7351 - f1: 0.7986

\*\*\*\*\*\*\* name results \*\*\*\*\*\*\*\*

acc: 0.8766 - recall: 0.8860 - f1: 0.8813

\*\*\*\*\*\*\* organization results \*\*\*\*\*\*\*\*

acc: 0.7512 - recall: 0.8311 - f1: 0.7891

\*\*\*\*\*\*\* position results \*\*\*\*\*\*\*\*

acc: 0.8182 - recall: 0.8106 - f1: 0.8144

\*\*\*\*\*\*\* scene results \*\*\*\*\*\*\*\*

acc: 0.7256 - recall: 0.7464 - f1: 0.7358

## 数据类别：

数据分为10个标签类别，分别为: 地址（address），书名（book），公司（company），游戏（game），政府（government），电影（movie），姓名（name），组织机构（organization），职位（position），景点（scene）

## 标签类别定义 & 标注规则：

地址（address）: \*\*省\*\*市\*\*区\*\*街\*\*号，\*\*路，\*\*街道，\*\*村等（如单独出现也标记）。地址是标记尽量完全的, 标记到最细。

书名（book）: 小说，杂志，习题集，教科书，教辅，地图册，食谱，书店里能买到的一类书籍，包含电子书。

公司（company）: \*\*公司，\*\*集团，\*\*银行（央行，中国人民银行除外，二者属于政府机构）, 如：新东方，包含新华网/中国军网等。

游戏（game）: 常见的游戏，注意有一些从小说，电视剧改编的游戏，要分析具体场景到底是不是游戏。

政府（government）: 包括中央行政机关和地方行政机关两级。 中央行政机关有国务院、国务院组成部门（包括各部、委员会、中国人民银行和审计署）、国务院直属机构（如海关、税务、工商、环保总局等），军队等。

电影（movie）: 电影，也包括拍的一些在电影院上映的纪录片，如果是根据书名改编成电影，要根据场景上下文着重区分下是电影名字还是书名。

姓名（name）: 一般指人名，也包括小说里面的人物，宋江，武松，郭靖，小说里面的人物绰号：及时雨，花和尚，著名人物的别称，通过这个别称能对应到某个具体人物。

组织机构（organization）: 篮球队，足球队，乐团，社团等，另外包含小说里面的帮派如：少林寺，丐帮，铁掌帮，武当，峨眉等。

职位（position）: 古时候的职称：巡抚，知州，国师等。现代的总经理，记者，总裁，艺术家，收藏家等。

景点（scene）: 常见旅游景点如：长沙公园，深圳动物园，海洋馆，植物园，黄河，长江等。

## 数据下载地址：

<https://www.cluebenchmarks.com/>

## 数据分布：

训练集：10748

验证集集：1343

按照不同标签类别统计，训练集数据分布如下（注：一条数据中出现的所有实体都进行标注，如果一条数据出现两个地址（address）实体，那么统计地址（address）类别数据的时候，算两条数据）：

【训练集】标签数据分布如下：

地址（address）:2829

书名（book）:1131

公司（company）:2897

游戏（game）:2325

政府（government）:1797

电影（movie）:1109

姓名（name）:3661

组织机构（organization）:3075

职位（position）:3052

景点（scene）:1462

【验证集】标签数据分布如下：

地址（address）:364

书名（book）:152

公司（company）:366

游戏（game）:287

政府（government）:244

电影（movie）:150

姓名（name）:451

组织机构（organization）:344

职位（position）:425

景点（scene）:199

## 数据字段解释：

以train.json为例，数据分为两列：text & label，其中text列代表文本，label列代表文本中出现的所有包含在10个类别中的实体。

例如：

text: "北京勘察设计协会副会长兼秘书长周荫如"

label: {"organization": {"北京勘察设计协会": [[0, 7]]}, "name": {"周荫如": [[15, 17]]}, "position": {"副会长": [[8, 10]], "秘书长": [[12, 14]]}}

其中，organization，name，position代表实体类别，

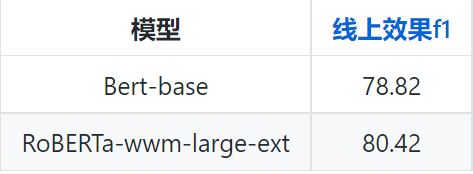
"organization": {"北京勘察设计协会": [[0, 7]]}：表示原text中，"北京勘察设计协会" 是类别为 "组织机构（organization）" 的实体, 并且start\_index为0，end\_index为7 （注：下标从0开始计数）

"name": {"周荫如": [[15, 17]]}：表示原text中，"周荫如" 是类别为 "姓名（name）" 的实体, 并且start\_index为15，end\_index为17

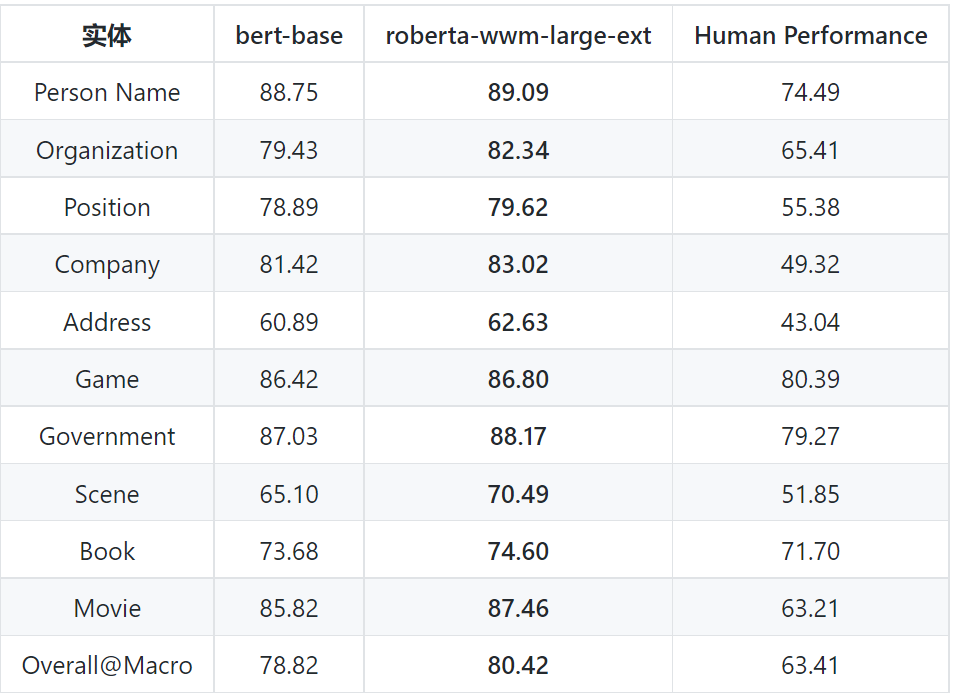
"position": {"副会长": [[8, 10]], "秘书长": [[12, 14]]}：表示原text中，"副会长" 是类别为 "职位（position）" 的实体, 并且start\_index为8，end\_index为10，同时，"秘书长" 也是类别为 "职位（position）" 的实体,

并且start\_index为12，end\_index为14

## 效果对比



各个实体的评测结果(F1 score)：



## 参考文献

[1] CLUENER2020: Fine-grained Name Entity Recognition for Chinese