NLP HW3 - 嵌套命名实体识别

丁豪 南京大学 人工智能学院
181220010@smail.nju.edu.cn

一、实验描述

本次实验为嵌套命名实体识别，即对于每个句子中可能存在嵌套性的明明实体进行抽取识别，并标注出其出现位置以及性质，不同实体之间的次序是没有关系的。本次实验的数据集由三部分给出，第一部分训练集由15022个句子构成，每个句子第一行为原始文本第二行为用 `|` 分隔开的不同命名实体，存在部分句子没有命名实体而空缺第二行的情况。第二部分开发集，由1669个句子组成，格式同训练集。第三部分测试集，共有1855个句子纯文本。实验允许使用外部词向量，但是不能使用如bert等语言模型，实验设置的满分阈值为f1=0.78。

二、运行方法

- 实验依赖文件已经描述在 **requirements.txt** 当中，执行 `pip install -r requirements.txt` 即可配置环境。
- 本实验依赖的**word embedding**的下载方式参见 **参考资料** 第二点，放到指定位置后可以进行之后操作。
- 实验的数据文件**train.txt dev.txt test.txt**需要放在 `./data/NLPHW/` 目录下。
- 本次实验使用 **jupyter notebook** 完成，建议在 **google-colab** 或同等GPU配置环境下运行以保障最佳性能。
    - 项目运行所需的文件有 `Nested.ipynb` 以及 `data,embeddings,model,module,reader,training,util` 文件夹下的全部内容（由于大小要求，embeddings与data并未上传，请按照上面两条进行准备后再运行）
    - 模型参数在第二个cell中指定，可以根据个人需要选择合适的参数进行训练

```
13  #  全局配置文件设置
14  class  Config:
15  >       def  __init__(self,dataset)  ->  None:···
70
71  >       def  __repr__(self)  ->  str:···
73
74  config  =  Config('NLPHW')
```

- 运行时直接点击 **jupyter notebook** 的 **Run All Cells** 按钮即可完成数据预处理、训练、预测、生成文件的完整过程。最终会在 `./dumps/` 目录下生成**log、模型、预测结果**txt三个文件。
- 生成预测结果后，会按照本次作业提交格式，在 `根目录` 下生成名为**181220010.txt**的提交文件

三、实现方法

- 数据预处理部分位于 `reader模块` ：将三个初始文本预处理为分词array，为每一个词设置相应id用于简单表示，最终加载预训练的word embedding产生每个词的word_vector。将以上文件用 `pickle` 存储于 `./data` 目录下。
- 使用pytorch实现的神经网络模块位于 `module` 中：主要有线性条件随机场（Linear-Chain-CRF），Drop-Out, 多层长短期记忆网络（MultyLayer-LSTM）三种组件。
- 组合使用上述模块构建模型位于 `model` 中：主体部分采用 `FastLSTM` 与DropOut的组合针对上述已经数组化的训练集进行训练，此外我们将CNN与CRF相结合得到的 `卷积条件随机场ConvCrf` 用于在保持条件独立性的情况下增大并行程度（主要体现在使用GPU的情况下）以加快训练速度。
- 训练过程的实现位于 `training`
- 其余一些工具类函数的定义在 `util`

四、实验结果

- 最终打榜f1：0.770421
- 训练时间在colab上Tesla T4显卡条件下 `一个epoch` 约为2分钟，本地Geforce 2060情况下约为3.5分钟，最终结果为训练 `60个epoch` 得到，由于固定了随机种子应当可以直接复现结果。
- 如需复现实验结果，请不要修改任何参数，并运行 `nested.ipynb` 。

五、参考资料

- 基本框架
  - 通过查阅**paperswithcode**网站，选取了**嵌套命名实体识别**任务中**不采用额外训练数据、GENIA数据集**上泛化性能较好且实现相对容易的**Second-best learning and decoding**

    https://paperswithcode.com/sota/nested-named-entity-recognition-on-genia
  - 由paperswithcode网站分享的github链接，采用了**Second-best learning and decoding**官方模型之一

    yahshibu/nested-ner-tacl2020-flair
- Word Embedding：按照上述github中的要求，下载并解压了PubMed-shuffle-win-2.bin，然后将其放到 `./embeddings` 文件夹中
- 此外，还需要感谢 181220031 李惟康 以及 181220056 王宸旭 同学在实验中与我进行的交流讨论，他们对于我理解实验内容以及开展实验有着积极的推动作用。

六、实验总结

通过本次实验，我对于嵌套命名实体识别这一任务有了一定的了解，对于其中的难点内容也有了一定的研究。同时通过使用和调试 `快速LSTM` 以及 `卷积条件随机场` ，我对当前业界解决嵌套命名实体识别这一任务的常用语言模型的基本原理和使用方法有了一些粗浅的认识。