基于 CRF 的药物副作用实体识别实验

2019/3/13 周开银

一、 实验环境配置(Linux or macOS)

1.1 配置虚拟环境

```
conda create -n python3.6 python=3.6 (创建虚拟环境 python3.6)
source activate python3.6 (打开虚拟环境)
source deactivate python3.6 (关闭虚拟环境)
Windows 系统请参考：https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html
```

注: 若 conda 命令不存在，请自行添加环境变量（Windows）

1.2 Wapiti 工具环境配置

```
source activate python3.6
mkdir yourProject (yourProject 是自定义的项目名)
cd yourProject
git clone https://github.com/Jekub/Wapiti.git
cd Wapiti
make
make install
./wapiti (有帮助文档输出表示安装成功)
```

注: 若 make 失败，请安装 gcc

1.3 工作环境配置

```
cd yourProject
git clone https://github.com/kyzhouhzau/2019SpringTextM.git
```

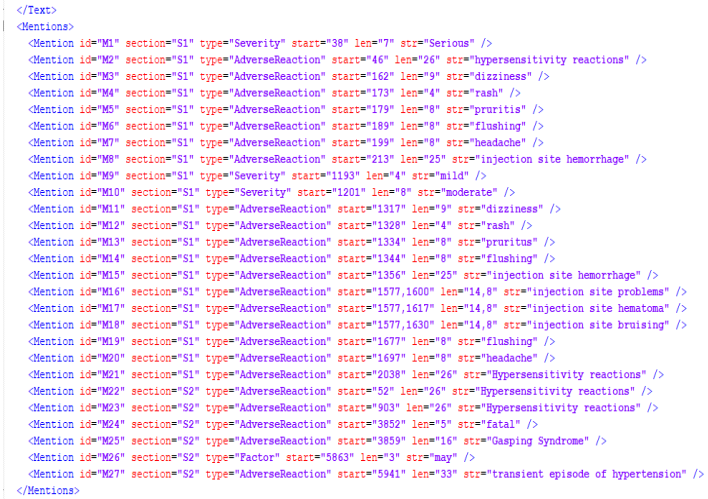
二、 数据描述

2.1 训练数据

训练数据有 100 个 Drug Label，格式如下面所展示。

Text:

Entity:



2.2 测试数据

从 DailyMed (https://dailymed.nlm.nih.gov) 中下载药物标签 34 个，格式为 XML。

（已经下载好在 2019SpringTextM 中）

例：

三、 数据处理

训练数据和测试数据的数据格式均为 XML，但不完全相同。在训练模型，和测试前均需要对数据进行预处理。将数据处理成".tab"格式。

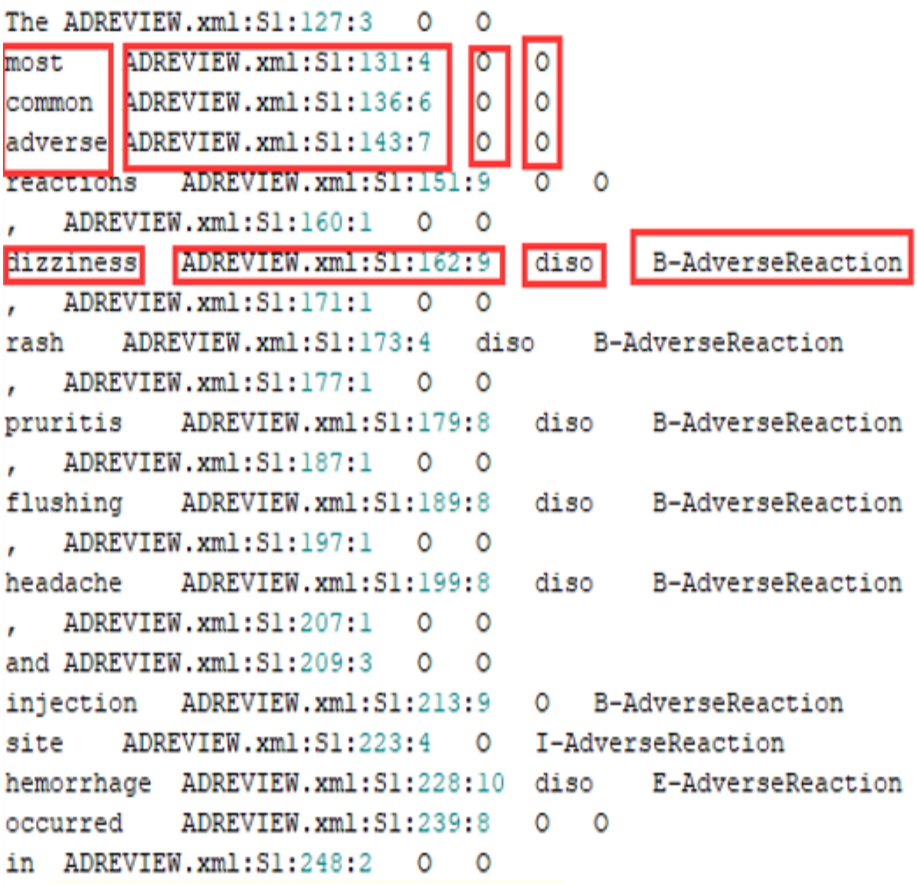
3.1 训练数据预处理

通过以下脚本预处理训练数据。

```
python tac2brat.py -d train_xml -o outtrain -F TokenDict:diso:diso-DISO.dic -t conll -s OBBEI
```

-d 训练文件夹
-o 预处理结果输出文件夹
-F 字典特征
-t 输出格式
-s 标签格式

输出格式：



注：同一横行中第一个红框中是分词后的单词，第二个红框是文件名：所属段落：起始位置：单词长度，第三个红框是字典特征，第四个红框是实体标签。在实体标签中对于某类标签如：标签 AdverseReaction 。若某个单词 “pruritis” 属于该标签则被标注为”B-AdverseReaction”;若某个词组“injection site hemorrhage”属于该标签，则该词组被标注为“B- AdverseReaction I-AdverseaeAction E-AdverseReaction”。于是，我们将该标签方式称为”BIEO”，此处 B-type 表示 begin, I-type 表示 Inside, E-type 表示 End, O 表示不属于该标签。

3.2 测试数据预处理

通过以下脚本处理测试数据。

```
cd 工作目录
python to_xml_needed.py 测试文件夹 output （注：在这里测试文件夹是 need_drug,表示我们从 dailyMed 下载的 XML 药物标签就放在该文件夹中。）
python tac2brat.py -d output -o outtest -F TokenDict:diso:diso-DISO.dic -t conll -s OBBEI
```

输出格式：

```
ALKERAN Alkeran.xml:S1:5:7  O   O
(   Alkeran.xml:S1:16:1 O   O
melphalan   Alkeran.xml:S1:17:9 O   O
hydrochloride   Alkeran.xml:S1:27:13    O   O
)   Alkeran.xml:S1:40:1 O   O
for Alkeran.xml:S1:43:3 O   O
Injection   Alkeran.xml:S1:47:9 O   O
Apo Alkeran.xml:S1:66:3 O   O
-   Alkeran.xml:S1:69:1 O   O
Pharma  Alkeran.xml:S1:70:6 O   O
USA Alkeran.xml:S1:77:3 O   O
,   Alkeran.xml:S1:80:1 O   O
Inc Alkeran.xml:S1:82:3 O   O
ALKERAN Alkeran.xml:S1:103:7    O   O
melphalan   Alkeran.xml:S1:114:9    O   O
hydrochloride   Alkeran.xml:S1:124:13   O   O
ALKERAN Alkeran.xml:S1:172:7    O   O
melphalan   Alkeran.xml:S1:183:9    O   O
hydrochloride   Alkeran.xml:S1:193:13   O   O
MELPHALAN   Alkeran.xml:S1:216:9    O   O
HYDROCHLORIDE   Alkeran.xml:S1:226:13   O   O
MELPHALAN   Alkeran.xml:S1:243:9    O   O
POVIDONES   Alkeran.xml:S1:260:9    O   O
DILUENT Alkeran.xml:S1:324:7    O   O
water   Alkeran.xml:S1:335:5    O   O
WATER   Alkeran.xml:S1:346:5    O   O
SODIUM  Alkeran.xml:S1:357:6    O   O
CITRATE Alkeran.xml:S1:364:7    O   O
PROPYLENE   Alkeran.xml:S1:377:9    O   O
GLYCOL  Alkeran.xml:S1:387:6    O   O
ALCOHOL Alkeran.xml:S1:399:7    O   O
WARNING Alkeran.xml:S1:485:7    O   O
Melphalan   Alkeran.xml:S1:494:9    O   O
should  Alkeran.xml:S1:504:6    O   O
be  Alkeran.xml:S1:511:2    O   O
administered    Alkeran.xml:S1:514:12   O   O
```

注：格式同训练数据

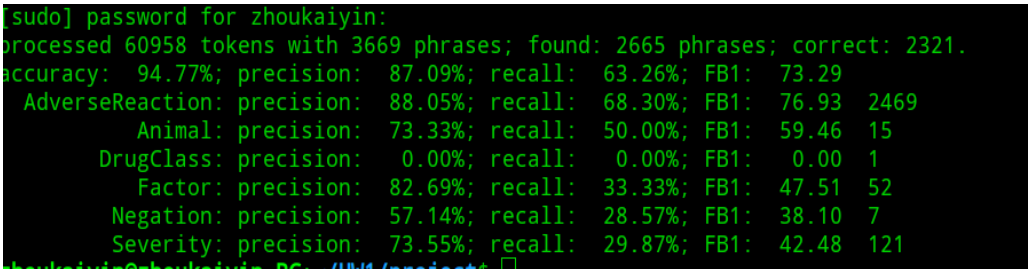
四、 模型训练

将训练数据随机按找 7:3 划分，在训练模型过程中，7 份用作实际训练模型，3 份用作开发集调整参数。当模型最优后用该参数训练所有 10 份数据获得模型，并对测试数据进行预测。

4.1 调整参数优化模型

```
sudo bash dev-wapiti.sh
```

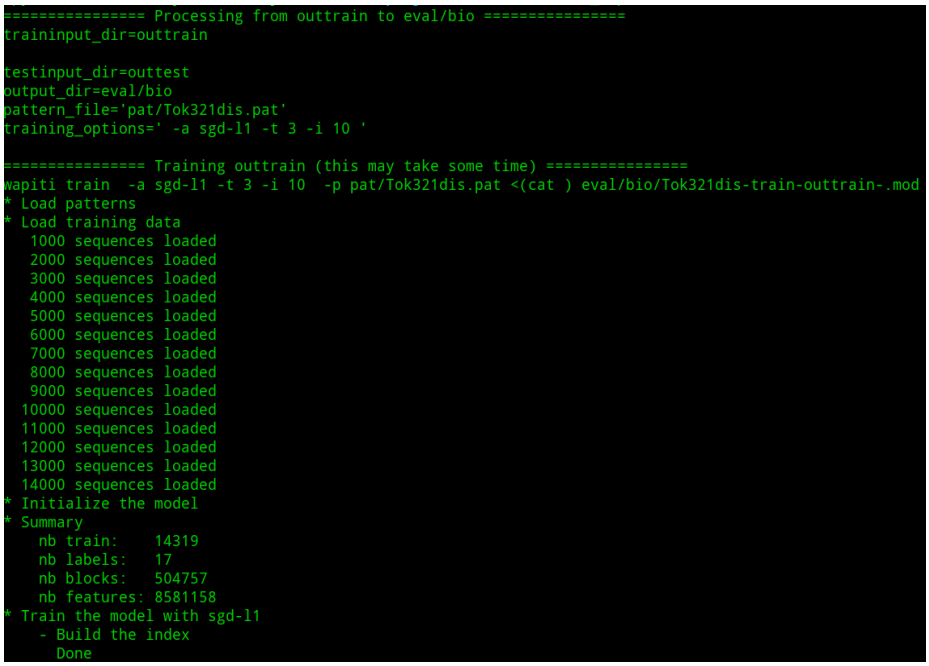
结果打印：



模型被存储在 eval/bio 中。

4.2 预测测试数据

```
sudo bash test_wapiti.sh
```



出现 Finished!说明序列标注完成。序列标注结果存储在 eval/bio/Tok321dis-train-test-outtrain.tab 中。