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:

```
More residently encoding with a content of southern that the content of the conte
```

Entity:

```
//Text>
(Mentions)
(Mention id="M1" section="S1" type="Severity" start="38" len="7" str="Serious" />
(Mention id="M2" section="S1" type="AdverseReaction" start="46" len="26" str="hypersensitivity reactions" />
(Mention id="M3" section="S1" type="AdverseReaction" start="173" len="4" str="cash" />
(Mention id="M3" section="S1" type="AdverseReaction" start="173" len="4" str="cash" />
(Mention id="M3" section="S1" type="AdverseReaction" start="173" len="8" str="pruritis" />
(Mention id="M3" section="S1" type="AdverseReaction" start="173" len="8" str="lensishing" />
(Mention id="M3" section="S1" type="AdverseReaction" start="193" len="8" str="lensishing" />
(Mention id="M3" section="S1" type="AdverseReaction" start="123" len="8" str="lensishing" />
(Mention id="M3" section="S1" type="AdverseReaction" start="123" len="8" str="lensishing" />
(Mention id="M3" section="S1" type="AdverseReaction" start="124" len="8" str="lensishing" />
(Mention id="M1" section="S1" type="AdverseReaction" start="134" len="8" str="lensishing" />
(Mention id="M12" section="S1" type="AdverseReaction" start="134" len="8" str="lensishing" />
(Mention id="M12" section="S1" type="AdverseReaction" start="134" len="8" str="lishing" />
(Mention id="M13" section="S1" type="AdverseReaction" start="134" len="8" str="lishing" />
(Mention id="M14" section="S1" type="AdverseReaction" start="134" len="8" str="lishing" />
(Mention id="M15" section="S1" type="AdverseReaction" start="134" len="8" str="lishing" />
(Mention id="M15" section="S1" type="AdverseReaction" start="135" len="8" str="lishing" />
(Mention id="M15" section="S1" type="AdverseReaction" start="135" len="8" str="lishing" />
(Mention id="M15" section="S1" type="AdverseReaction" start="135" len="8" str="lishing" />
(Mention id="M25" section="S1" type="AdverseReaction" start="135" len="8" str="lishing" />
(Mention id="M25" section="S1" type="AdverseReaction" start="135" len="8" str="lishing" />
(Mention id="M25" section="S2" type="AdverseReaction" start="135" len="8" str="lishing" />
(Men
```

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:

```
The ADREVIEW.xml:S1:127:3
most
       ADREVIEW.xml:S1:131:4
       ADREVIEW.xml:S1:136:6
                             0
common
adverse ADREVIEW.xml:S1:143:7
                            0
reactions ADREVIEW.xml:S1:151:9
   ADREVIEW.xml:S1:160:1 O
dizziness ADREVIEW.xml:S1:162:9
                                        B-AdverseReaction
                                diso
   ADREVIEW.xml:S1:171:1 O
rash
       ADREVIEW.xml:S1:173:4 diso
                                    B-AdverseReaction
   ADREVIEW.xml:S1:177:1 0 0
pruritis
          ADREVIEW.xml:S1:179:8 diso
                                      B-AdverseReaction
   ADREVIEW.xml:S1:187:1 O O
flushing
          ADREVIEW.xml:S1:189:8
                                diso B-AdverseReaction
   ADREVIEW.xml:S1:197:1 0 0
headache
         ADREVIEW.xml:S1:199:8 diso B-AdverseReaction
   ADREVIEW.xml:S1:207:1 0 0
and ADREVIEW.xml:S1:209:3 O O
injection ADREVIEW.xml:S1:213:9 O B-AdverseReaction
     ADREVIEW.xml:S1:223:4 O I-AdverseReaction
hemorrhage ADREVIEW.xml:S1:228:10 diso E-AdverseReaction
          ADREVIEW.xml:S1:239:8 O O
occurred
in ADREVIEW.xml:S1:248:2 O O
```

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-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 0 0
melphalan Alkeran.xml:S1:17:9 0 0
hydrochloride Alkeran.xml:S1:27:13
   Alkeran.xml:S1:40:1 O
                         0
for Alkeran.xml:S1:43:3 O
                         0
         Alkeran.xml:S1:47:9 O
Injection
Apo Alkeran.xml:S1:66:3 0
   Alkeran.xml:S1:69:1 O
Pharma Alkeran.xml:S1:70:6 O
USA Alkeran.xml:S1:77:3 0 0
   Alkeran.xml:S1:80:1 O
                         0
Inc Alkeran.xml:S1:82:3 O
ALKERAN Alkeran.xml:S1:103:7
melphalan Alkeran.xml:S1:114:9
hydrochloride
              Alkeran.xml:S1:124:13
ALKERAN Alkeran.xml:S1:172:7 O
melphalan Alkeran.xml:S1:183:9
                                0
hvdrochloride Alkeran.xml:S1:193:13
                                   0
MELPHALAN Alkeran.xml:S1:216:9 0
HYDROCHLORIDE
              Alkeran.xml:S1:226:13
MELPHALAN Alkeran.xml:S1:243:9 O
POVIDONES
          Alkeran.xml:S1:260:9
                                 O
DILUENT Alkeran.xml:S1:324:7
                                 0
water Alkeran.xml:S1:335:5
       Alkeran.xml:S1:346:5
SODIUM Alkeran.xml:S1:357:6
                             0
                                 0
CITRATE Alkeran.xml:S1:364:7
                             0
                                 0
PROPYLENE Alkeran.xml:S1:377:9
                                 0
                                     0
GLYCOL Alkeran.xml:S1:387:6
                            0
                                 0
ALCOHOL Alkeran.xml:S1:399:7
                             0
                                 0
WARNING Alkeran.xml:S1:485:7
                             0
                                 0
Melphalan Alkeran.xml:S1:494:9
                                0
should Alkeran.xml:S1:504:6
                            0
                                 O
be Alkeran.xml:S1:511:2 O
             Alkeran.xml:S1:514:12 O
administered
```

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:

```
[sudo] password for zhoukaiyin:

processed 60958 tokens with 3669 phrases; found: 2665 phrases; correct: 2321.

accuracy: 94.77%; precision: 87.09%; recall: 63.26%; FB1: 73.29

AdverseReaction: precision: 88.05%; recall: 68.30%; FB1: 76.93 2469

Animal: precision: 73.33%; recall: 50.00%; FB1: 59.46 15

DrugClass: precision: 0.00%; recall: 0.00%; FB1: 0.00 1

Factor: precision: 82.69%; recall: 33.33%; FB1: 47.51 52

Negation: precision: 57.14%; recall: 28.57%; FB1: 38.10 7

Severity: precision: 73.55%; recall: 29.87%; FB1: 42.48 121
```

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。<mark>序列标注结果</mark>存储在 eval/bio/Tok321dis-train-test-outtrain.tab 中。