作业五.Wapiti 序列标注的 CRF 实现和.pat 模板测试

2021 年 5 月 26 日

学　　校:	华中农业大学
学院班级:	信息学院生信 1801 班
姓　　名:	邓启东
学　　号:	2018317220103
指导教师:	夏静波

目录

1 实验目的

2 实验材料与方法

2.1 CRF 算法介绍

条件随机域模型是由 Lafferty 在 2001 年提出的一种典型的判别式模型。它在观测序列的基础上对目标序列进行建模, 重点解决序列化标注的问题。条件随机域模型既具有判别式模型的优点, 又具有产生式模型考虑到上下文标记间的转移概率, 以序列化形式进行全局参数优化和解码的特点, 解决了其他判别式模型 (如最大熵马尔科夫模型) 难以避免的标记偏置问题。

2.2 wapiti 工作环境配置

wapiti 是一个实现 CRF 算法的工具安装好就编译成为了二进制文件解压整个项目压缩文件，并解压其中的 AGAC 语料库，其中有训练集和测试集。使用 unrar 解压失败 (见图1)。

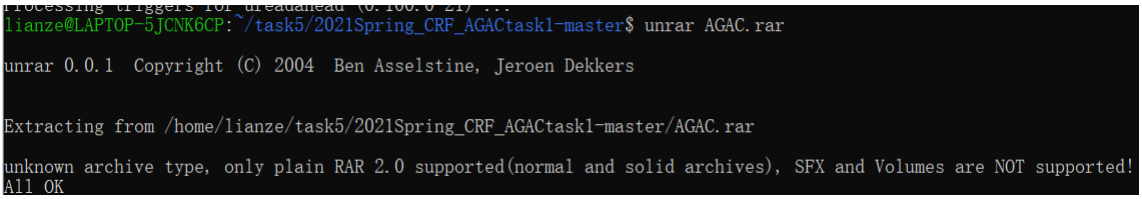

图 1: 未知类型：unrar 无法解压

改换 7zip，使用代码 sudo apt-get install p7zip-full p7zip-rar 安装，代码 7z x AGAC.rar 解压。解压成功。

2.3 数据描述

实际上，原本的 AGAC 语料库文件我们在任务一中是见到过的，原本是一个 jason 格式的文件。jason 文件是以字典的形式储存 text 和 denotations 的。经过 Python 的 nltk 包处理之后，将实体标签和分词根据根据未知信息对应，就得到了相应形式的 BIO 文件，作为我们的训练集。

我们随便打开一篇，例如打开/AGAC/test_split/ddd_split_211.txt 这一篇。第一列便是我们的 input，其实就是序列标注的句子，每一行就是句子的一个单词。而它的实际标签就是最后一列，以 BIO 的形式呈现，例如 B-Var、B-Gene、B-Enzyme、I-MPA 等等。以这种方式就可以很好的对多个单词组成的短语进行标注 (见图2)。

这里明显看到 Denys-Drash syndrome 这个疾病的边界被很好的标注出来了。

```
Syndrome        28081536        0       0       diso:dicT047    0
.       28081536        0       0       0       O

Denys   28081536        0       0       0       B-Disease
-       28081536        0       0       0       I-Disease
Drash   28081536        0       0       0       I-Disease
syndrome        28081536        0       0       diso:dicT047    I-Disease
(       28081536        0       0       0       O
DDS     28081536        0       0       diso:dicT047    0
)       28081536        0       0       0       O
is      28081536        0       0       0       O
characterized   28081536        0       0       0       O
by      28081536        0       0       0       O
nephropathy     28081536        0       0       diso:dicT047    0
,       28081536        0       0       0       O
```

图 2: 标注格式

这里的训练集包含 500 篇摘要，由标注员手工标注的。由语料库开发到黄金数据库花费了 20 个月。但是也意味着这个训练集非常的准确。

第一列语料库文件中的内容分割后得到的，包括单词，数字和标点符号

第二列的数字是 PMID

第五列的 diso 是自己造的词典，收纳了和不良反应的词，表示 diso 中的副作用，这个词典在 diso-DISO.dic 文件可以看到。

而测试集的标签全部为 O。

2.4 训练模型

wapiti train -a sgd-l1 -t 3 -i 10 -p pat/Tok321dis.pat <(cat AGAC/train_split/*.txt) AGAC/mod/AGAC_train.mod

这句代码其实就是把 AGAC/train_split/路径下所有的语料库文件数据用于训练。

“<”是指将这个训练集进去。

-p 指定模板文件，只有 wapiti 这么提，其实是特征函数的生成方式，Tok321dis.pat 中就是特征函数。

我们打开这个文件, 见图3。

```
lianze@LAPTOP-5JCNK6CP:~/task5/2021Spring_CRF_AGACtask1-master/pat$ cat Tok321dis.pat
*

U:tok:1:-1:%X[-1,0]
U:tok:1:+0:%X[0,0]
U:tok:1:+1:%X[1,0]

U:tok:2:-1:%X[-1,0]/%X[0,0]
U:tok:2:+0:%X[0,0]/%X[1,0]

U:tok:3:-2:%X[-2,0]/%X[-1,0]/%X[0,0]
U:tok:3:-1:%X[-1,0]/%X[0,0]/%X[1,0]
U:tok:3:+0:%X[0,0]/%X[1,0]/%X[2,0]


U:pre:1:+0:4:%M[0,0,"^.?.?.?.?"]

U:suf:1:+0:4:%M[0,0,".?.?.?.?$"]

U:dis:1:-1:%X[-1,2]
U:dis:1:+0:%X[0,2]
U:dis:1:+1:%X[1,2]
```

图 3: 特征函数生成模板

训练出来的模型便保存在了 AGAC_train.mod 中。由于开头第一个字母是 U, 这样相对于输出序列而言的表示方式是 Unigram 模板，意思是前一个数据，后一个数据和当前的数据构成的上下文对当前标签的影响，还有一种表示方式是 Bigram 模板，这个模板会考虑当前输出标签和上一个输出标签。

模板文件中的每一行代表一个 template。每一个 template 中，专门的宏%x[row,col] 用于确定输入数据中的一个 token。row 用于确定与当前的 token 的相对行数。col 用于确定列。

结果如下图所示（见图4）。

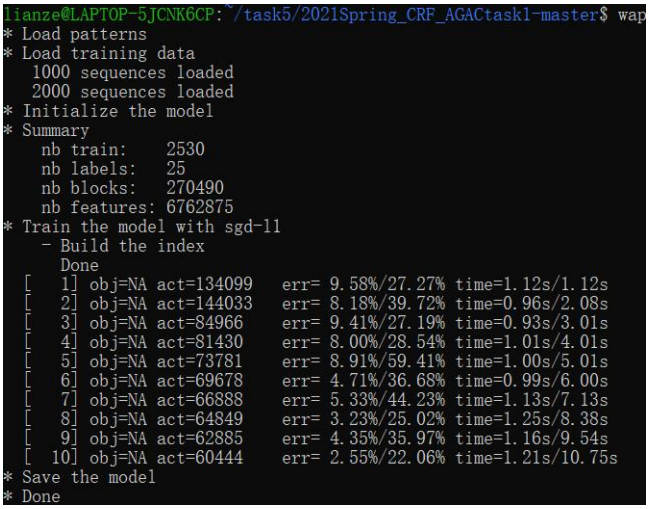

图 4: pretrain

可以看到由特征模板学到的特征有 6,762,875。

2.5 预测标签

wapiti label -c -m AGAC/mod/AGAC_train.mod <(cat AGAC/test_split/*.txt) AGAC/train_out.tab

也就是拿训练好的模型来预测 test_split 目录下的那 250 篇文献，如下图，可以看到每一个标签的预测情况（见图5）。如果打开 train_out.tab, 可以直接看到第五列后面多了一列预测。可以和实际的结果相互比较。

```
lianze@LAPTOP-5JCNK6CP:~/task5/2021Spring_CRF_AGACtask1-master$ wapiti label -c -m AGAC/mod/AGAC_train.mod <(cat AGAC/test_split/*.txt) AGAC/train_out.tab
* Load model
* Label sequences
     1000 sequences labeled     8.19%/34.30%
     2000 sequences labeled     8.04%/33.95%
   Nb sequences  : 2506
   Token error   :  7.88%
   Sequence error: 33.64%
* Per label statistics
   O          Pr=0.94  Rc=0.98  F1=0.96
   B-Var      Pr=0.45  Rc=0.28  F1=0.34
   B-Gene     Pr=0.52  Rc=0.03  F1=0.05
   B-Enzyme   Pr=0.00  Rc=0.00  F1=-nan
   B-PosReg   Pr=0.32  Rc=0.23  F1=0.27
   B-MPA      Pr=0.17  Rc=0.04  F1=0.07
   I-MPA      Pr=0.13  Rc=0.02  F1=0.04
   B-Disease  Pr=0.28  Rc=0.12  F1=0.16
   I-Disease  Pr=0.40  Rc=0.15  F1=0.22
   I-Var      Pr=0.26  Rc=0.25  F1=0.25
   B-Interaction  Pr=0.17  Rc=0.14  F1=0.15
   B-Protein  Pr=-nan  Rc=0.00  F1=-nan
   B-Pathway  Pr=-nan  Rc=0.00  F1=-nan
   I-Pathway  Pr=-nan  Rc=0.00  F1=-nan
   B-NegReg   Pr=0.71  Rc=0.23  F1=0.35
   B-Reg      Pr=0.54  Rc=0.05  F1=0.09
   I-PosReg   Pr=0.61  Rc=0.83  F1=0.70
   B-CPA      Pr=0.22  Rc=0.09  F1=0.13
   I-CPA      Pr=0.21  Rc=0.03  F1=0.06
   I-NegReg   Pr=1.00  Rc=0.46  F1=0.63
   I-Gene     Pr=0.77  Rc=0.07  F1=0.12
   I-Reg      Pr=0.69  Rc=0.06  F1=0.11
   I-Protein  Pr=-nan  Rc=0.00  F1=-nan
   I-Interaction  Pr=-nan  Rc=0.00  F1=-nan
   I-Enzyme   Pr=0.00  Rc=0.00  F1=-nan
* Done
```

图 5: 标签预测完成

O 本身就占有大部分，所以准确度和召回率都很高自不必说。

2.6 评估结果

perl conlleval.pl -d $'\ t'<AGAC/train_out.tab | tee AGAC/train_out.eval
评估结果见下图（图/refresult）

```
lianze@LAPTOP-5JCNK6CP:~/task5/2021Spring_CRF_AGACtask1-master$ perl conlleval.pl -d $'\t' <AGAC/train_out.tab | tee AGAC/train_out.eval
processed 62559 tokens with 2416 phrases; found: 781 phrases; correct: 279.
accuracy:  92.12%; precision:  35.72%; recall:  11.55%; FB1:  17.45
               CPA: precision:  21.74%; recall:   8.93%; FB1:  12.66  23
           Disease: precision:  22.41%; recall:   9.33%; FB1:  13.18  174
            Enzyme: precision:   0.00%; recall:   0.00%; FB1:   0.00  2
              Gene: precision:  52.17%; recall:   2.58%; FB1:   4.91  23
       Interaction: precision:  16.67%; recall:  14.29%; FB1:  15.38  6
               MPA: precision:  14.81%; recall:   3.98%; FB1:   6.27  54
            NegReg: precision:  71.43%; recall:  23.44%; FB1:  35.29  42
           Pathway: precision:   0.00%; recall:   0.00%; FB1:   0.00  0
            PosReg: precision:  31.67%; recall:  23.17%; FB1:  26.76  60
           Protein: precision:   0.00%; recall:   0.00%; FB1:   0.00  0
               Reg: precision:  51.35%; recall:   4.77%; FB1:   8.74  37
               Var: precision:  40.56%; recall:  25.75%; FB1:  31.50  360
```

图 6: 预测结果评估

3 .pat 模板修改测试

3.1 .pat 模板修改

为了提高结果的 FB1 的值，则需要修改.pat 文件里面的特征模板，特征的生成方式变得更加复杂，tok 可以提高到 4。

修改后的.pat 文件如下图所示（见下图7）：

```
*

U:tok:1:-1:%X[-1,0]
U:tok:1:+0:%X[0,0]
U:tok:1:+1:%X[1,0]

U:tok:2:-1:%X[-1,0]/%X[0,0]
U:tok:2:+0:%X[0,0]/%X[1,0]

U:tok:3:-2:%X[-2,0]/%X[-1,0]/%X[0,0]
U:tok:3:-1:%X[-1,0]/%X[0,0]/%X[1,0]
U:tok:3:+0:%X[0,0]/%X[1,0]/%X[2,0]

U:tok:4:-3:%X[-3,0]/%X[-2,0]/%X[-1,0]/%X[0,0]
U:tok:4:-2:%X[-2,0]/%X[-1,0]/%X[0,0]/%X[1,0]
U:tok:4:-1:%X[-1,0]/%X[0,0]/%X[1,0]/%X[2,0]
U:tok:4:+0:%X[0,0]/%X[1,0]/%X[2,0]/%X[3,0]

U:pre:1:+0:4:%M[0,0,"^.?.?.?.?"]

U:suf:1:+0:4:%M[0,0,".?.?.?.?$"]

U:dis:1:-1:%X[-1,2]
U:dis:1:+0:%X[0,2]
U:dis:1:+1:%X[1,2]
```

图 7: 修改后的 pat 文件

3.2 修改后结果

这个时候我们的特征已经有 12，922，075 个了。但是可能是由于特征太多的原因，导致了学习的效果不增反降（见下图8）。

```
lianze@LAPTOP-5JCNK6CP:~/task5/2021Spring_CRF_AGACtask1-master$ perl conlleval.pl -d $'\t' <AGAC/train_out.tab | tee A
C/train_out.eval
processed 62559 tokens with 2416 phrases; found: 858 phrases; correct: 254.
accuracy:  91.72%; precision:  29.60%; recall:  10.51%; FB1:  15.52
             CPA: precision:   9.76%; recall:   7.14%; FB1:   8.25  41
         Disease: precision:  25.20%; recall:  15.07%; FB1:  18.86  250
          Enzyme: precision:   0.00%; recall:   0.00%; FB1:   0.00  4
            Gene: precision:  30.65%; recall:   4.08%; FB1:   7.20  62
     Interaction: precision:   0.00%; recall:   0.00%; FB1:   0.00  5
             MPA: precision:  18.60%; recall:   3.98%; FB1:   6.56  43
          NegReg: precision:  67.57%; recall:  19.53%; FB1:  30.30  37
         Pathway: precision:   0.00%; recall:   0.00%; FB1:   0.00  0
          PosReg: precision:  35.90%; recall:  34.15%; FB1:  35.00  78
         Protein: precision:   0.00%; recall:   0.00%; FB1:   0.00  2
             Reg: precision:  68.18%; recall:   3.77%; FB1:   7.14  22
             Var: precision:  29.30%; recall:  16.23%; FB1:  20.89  314
```

图 8: 预测结果评估

分析原因，估计是语料库还不够大，特征过多导致预测效果不佳。不过我们此刻还没有放弃，因为 FB1 的值不增反降。因此我打算插入 tok5，查看结果。如果按照理论而言，如果将模板扩增到 tok4，那么特征的数量就会多更多更多。如果效果更差，也许特征太多的原因。如果 tok5 的 FB1 的值最大，那么说明 tok4 效果不佳，具体原因不太明白。

3.3 增加到 tok5

我们把 U 模板扩增到 tok5（见下图9）。

```
19 U:tok:5:-4:%X[-4,0]/%X[-3,0]/%X[-2,0]/%X[-1,0]/%X[0,0]
20 U:tok:5:-3:%X[-3,0]/%X[-2,0]/%X[-1,0]/%X[0,0]/%X[1,0]
21 U:tok:5:-2:%X[-2,0]/%X[-1,0]/%X[0,0]/%X[1,0]/%X[2,0]
22 U:tok:5:-1:%X[-1,0]/%X[0,0]/%X[1,0]/%X[2,0]/%X[3,0]
23 U:tok:5:+0:%X[0,0]/%X[1,0]/%X[2,0]/%X[3,0]/%X[4,0]
```

图 9: 增加到 tok5

可以明显感受到整个训练、预测、评估的过程所花费的时间长了很多。结果如下图所示（10）。

```
lianze@LAPTOP-5JCNK6CP:~/task5/2021Spring_CRF_AGACtask1-master$ perl conlleval.pl -d
C/train_out.eval
processed 62559 tokens with 2416 phrases; found: 1091 phrases; correct: 349.
accuracy:  91.64%; precision:  31.99%; recall:  14.45%; FB1:  19.90
              CPA: precision:  16.00%; recall:   7.14%; FB1:   9.88  25
          Disease: precision:  23.90%; recall:  15.55%; FB1:  18.84  272
           Enzyme: precision:   0.00%; recall:   0.00%; FB1:   0.00  1
             Gene: precision:  26.47%; recall:   3.86%; FB1:   6.74  68
      Interaction: precision:  10.00%; recall:  14.29%; FB1:  11.76  10
              MPA: precision:  15.94%; recall:   5.47%; FB1:   8.15  69
           NegReg: precision:  56.45%; recall:  27.34%; FB1:  36.84  62
          Pathway: precision:   0.00%; recall:   0.00%; FB1:   0.00  1
           PosReg: precision:  35.71%; recall:  24.39%; FB1:  28.99  56
          Protein: precision:   0.00%; recall:   0.00%; FB1:   0.00  3
              Reg: precision:  54.05%; recall:   5.03%; FB1:   9.20  37
              Var: precision:  35.93%; recall:  30.86%; FB1:  33.21  487
```

图 10: 增加到 tok5

这次看到，FB1 的值有明显提高，比 tok3 整体要好很多。不过部分预测能力也不如 tok3，比如对于 CPA 的识别，tok3 的 FB1 值是 12.66 而 tok5 仅有 9.88。Interaction 这一栏，tok3 的 FB1 值是 15.38,tok5 是 11.76. 不过这些都比 tok4 的 Interaction 准确率，召回率，FB1 均为 0 要好。

这说明 tok4 本身学到的特征还是有些许问题的。也说明虽然在个别标签反有不足，总体来说增加模板的上下文词数来增加特征函数还是一定程度上可以改进标签预测的效果的。

4 参考链接

1. 原项目 Github 链接
2. 参考博客