

零基础入门--中文命名实体识别

原创 buppt BUPPT 2020-03-14

中文分词

说到命名实体抽取，先要了解一下基于字标注的中文分词。比如一句话

"我爱北京天安门"。

分词的结果可以是

"我/爱/北京/天安门"。

那什么是基于字标注呢？

"我/O 爱/O 北/B 京/E 天/B 安/M 门/E"。

就是这样，给每个字都进行一个标注。我们可以发现这句话中字的标注一共有四种。他们分别代表的意义如下。

B | 词首

M | 词中

E | 词尾

O | 单字

B 表示一个词的开始，E 表示一个词的结尾，M 表示词中间的字。如果这个词只有一个字的话，用 O 表示。

命名实体识别

数据处理

了解了中文分词，那么实体识别也差不多。就是把不属于实体的字用 O 标注，把实体用 BME 规则标注，最后按照 BME 规则把实体提取出来就 ok 了。

数据可以自己标注，也可以找个公开的数据集先练练手。我用的是玻森数据提供的命名实体识别数据，<https://bosonnlp.com> 这是官网，在数据下载里面有一个命名实体识别数据集，或者在我的 github 里下载。

这个数据集一个包含了 6 个实体类别：

```
time: 时间
location: 地点
personname: 人名
orgname: 组织名
companyname: 公司名
productname: 产品名
```

例：

```
{productname:浙江在线杭州}{time:4 月 25 日}讯 (记者{personname: 施宇翔} 通讯员
{personname:方英}) 毒贩很 “时髦” , 用{productname:微信}交易毒品。没料想警方也很
“潮” , 将计就计, 一举将其擒获。
```

每个实体都用大括号括了起来，并标明实体类别。当然自己标注的时候也不一定要这么标，只要能提取出来就可以。

然后我们要做的就是将原始数据按照 BMEO 规则变成字标注的形式，以便模型训练。这使用 python 实现还是比较简单的，嫌麻烦的可以看我的 github 里的代码。按字标注后结果如下。

```
浙/Bproductname 江/Mproductname 在/Mproductname 线/Mproductname
杭/Mproductname 州/Eproductname 4/Btime 月/Mtime 2/Mtime 5/Mtime 日/Etime 讯/O
(/O 记/O 者/O /Bpersonname 施/Mpersonname 宇/Mpersonname 翔/Epersonname /O
通/O 讯/O 员/O /O 方/Bpersonname 英/Epersonname ) /O 毒/O 贩/O 很/O “/O 时/O 髦/O
” /O , /O 用/O 微/Bproductname 信/Eproduct_name 交/O 易/O 毒/O 品/O 。 /O 没/O 料/O
想/O 警/O 方/O 也/O 很/O “/O 潮/O ” /O , /O 将/O 计/O 就/O 计/O , /O 一/O 举/O 将/O
其/O 擒/O 获/O 。
```

然后我们习惯按照标点符号把一个长句分成几个短句，反正一般实体里面也没有标点符号。结果如下。

浙/Bproductname 江/Mproductname 在/Mproductname 线/Mproductname
 杭/Mproductname 州/Eproductname 4/Btime 月/Mtime 2/Mtime 5/Mtime 日/Etime 讯/O
 记/O 者/O /Bpersonname 施/Mpersonname 宇/Mpersonname 翔/Epersonname /O 通/O
 讯/O 员/O /O 方/Bpersonname 英/Epersonname
 毒/O 贩/O 很/O
 时/O 髦/O
 用/O 微/Bproductname 信/Eproduct_name 交/O 易/O 毒/O 品/O
 没/O 料/O 想/O 警/O 方/O 也/O 很/O
 潮/O
 将/O 计/O 就/O 计/O
 一/O 举/O 将/O 其/O 擒/O 获/O

然后的思路就是建立一个 word2id 词典，把每个汉字转换成 id。这里习惯性按照数据集中每个汉字出现的次数排序，id 从 1 开始。

```
16753
16753
的      1
1       2
0       3
        4
2       5
在      6
中      7
国      8
年      9
一     10
了     11
日     12
是     13
月     14
大     15
为     16
人     17
3     18
上     19
有     20
5     21
行     22
```

 <https://blog.csdn.net/buppt>

再建立一个 tag2id 词典，把每一个字标注的类型转换成 id。这里的顺序我就随便搞的。

```

吐      3433
刃      3434
dtype: int64

0
B_product_name      1
B_org_name          2
M_org_name          3
E_time             4
M_product_name      5
M_person_name       6
E_location          7
M_time             8
0                  9
E_product_name     10
B_time            11
E_person_name     12
E_company_name    13
M_company_name    14
M_location        15
B_company_name    16
B_person_name     17
B_location        18
E_org_name        19
dtype: int64

```


 <https://blog.csdn.net/buppt>

之后就把刚按标点分开的数据，按照一一对应的顺序，把汉字和每个字的标签转换成 id，分别存到两个数组里面，一起保存到一个 pkl 文件中，这样模型使用时就可以直接读取，不用每次都处理数据了。这里习惯把每一句话都转换成一样的长度。这个长度当然是自己设置的，比它长的就把后面舍弃，比它短的就在后面补零。

```

[529 192  6 313 521 156  32  14  5  21  12 184  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0]
[ 1  5  5  5  5 10 11  8  8  8  4  9  0  0  0  0  0  0
  0  0  0  0  0  0]

```

 <https://blog.csdn.net/buppt>

这里第一个数组里是这句话汉字转换成的 id，第二个数组里存的是这句话每个字的标注转换成的 id。

训练

我的 github 里有两个版本，pytorch 版直接用的 pytorch tutorial 里的 Bilstm+crf 模型。

运行 train.py 训练即可。由于使用的是 cpu，而且也没有使用 batch，所以训练速度比较慢。想简单跑一下代码的话，建议只使用部分数据跑一下。pytorch 暂时不再更新。

开始训练

使用 `python train.py` 开始训练，训练的模型会存到 model 文件夹中。


使用预训练的词向量

使用 `python train.py pretrained` 会使用预训练的词向量开始训练, `vec.txt` 是在网上找的一个比较小的预训练词向量, 可以参照我的代码修改使用其他更好的预训练词向量。

测试训练好的模型

使用 `python train.py test` 进行测试, 会自动读取 `model` 文件夹中最新的模型, 输入中文测试即可, 测试结果好坏根据模型的准确度而定。

```
loading pre-trained model from ./model/model20.ckpt.....
Enter your input: 我在即将结束对美国国事访问之际, 来到美国西海岸第一大城市洛杉矶, 感到由衷的高兴。
result:
ns:美国
ns:美国
ns:洛杉矶
Enter your input: 我要感谢洛杉矶市民议政论坛、亚洲协会南加中心、美中关系全国委员会、美中友协美西分会等友好团体的盛情款待。
result:
nr:洛杉矶
nt:亚洲协会南加中心
nt:美中友协美西分会
Enter your input: 在华盛顿期间, 我同克林顿总统就双边关系以及共同关心的国际和地区问题交换了意见, 取得了积极的、建设性的成果。
result:
ns:华盛顿
nr:克林顿
Enter your input: 
```

<https://blog.csdn.net/buppt>  BUPPT


文件级别实体抽取

使用 `python train.py input_file output_file` 进行文件级实体抽取。

可以自动读取 `model` 文件夹中最新的模型, 将 `input_file` 中的实体抽取出来写入 `output_file` 中。先是原句, 然后是实体类型及实体 (可按照需要修改)。

如 `python train.py test1.txt res.txt`, `res.txt` 内容如下:

```
江主席在洛杉矶国际机场发表了告别讲话    ns:洛杉矶国际机场
全文另发
对克林顿总统和夫人以及美国政府和人民给予的热情和隆重的接待表示谢意    nr:克林顿 nt:美国政府
洛杉矶市市长理查德·赖尔登为江主席举行了隆重的欢送仪式    ns:洛杉矶市 nr:查德·赖尔登为江
到机场送行的美方人员有: 加利福尼亚州长代表邝杰灵    ns:加利福尼亚
美国驻华大使尚慕杰等    ns:美国 nr:尚慕杰
中国驻美大使李道豫    ns:中国 nr:李道豫
外交部副部长李肇星    nt:外交部 nr:李肇星
中国驻洛杉矶总领事冯树森以及中国驻美使    ns:中国 ns:洛杉矶 nr:冯树森 ns:中国
领馆的工作人员和当地华人
华侨
留学生代表也到机场欢送江主席一行
```

 BUPPT

准确度判断

命名实体识别的准确度判断有三个值。准确率、召回率和 f 值。

这里需要先定义一个交集，是经过模型抽取出来的实体，与数据集中的所有实体，取交集。

准确率=交集/模型抽取出的实体

召回率=交集/数据集中的所有实体

f 值= $2 \times (\text{准确率} \times \text{召回率}) / (\text{准确率} + \text{召回率})$

喜欢此内容的人还喜欢

【岛妹说】特斯拉风波，不只是维权的事

侠客岛

中国放贷者败走印度：大量公司账户遭冻结，部分印度人称中国人傻钱多

腾讯财经