# [CRF++词性标注](http://x-algo.cn/index.php/2016/02/28/crf-tagging/)

2016-02-28 分类：[NLP](http://x-algo.cn/index.php/category/nlp/) 阅读(7915) 评论(20)

训练和测试的语料都是人民日报98年标注语料，训练和测试比例是10：1，直接通过CRF++标注词性的准确率:0.933882。特征有一千多万个，训练时间比较长。机器cpu是48核，通过crf++，指定并线数量 -p为40，训练了大概七个小时才结束。

语料库、生成训练数据的python脚本、训练日志、模型、计算准确率脚本都上传到网盘，可以直接下载：[戳我下载 CRF++词性标注](http://pan.baidu.com/s/1bokPQLx" \t "http://x-algo.cn/index.php/2016/02/28/crf-tagging/_blank)，程序在centos6.5+python2.7下面运行通过，如果在win下或者ubuntu下可能会有异常，通常都是编码、路径规范等小问题，通过逐行debug脚本应该很容易找到问题，同时要确定crf++在自己机器本身编译没有问题，下面说一下每一步的过程。

**文章目录**[[隐藏](http://x-algo.cn/index.php/2016/02/28/crf-tagging/)]

* [1 生成训练和测试数据](http://x-algo.cn/index.php/2016/02/28/crf-tagging/" \l "i)
* [2 执行训练和测试](http://x-algo.cn/index.php/2016/02/28/crf-tagging/" \l "i-2)
* [3 计算准确率](http://x-algo.cn/index.php/2016/02/28/crf-tagging/" \l "i-3)
* [4 实验结果](http://x-algo.cn/index.php/2016/02/28/crf-tagging/" \l "i-4)

## **生成训练和测试数据**

生成训练和测试数据脚本：get\_post\_train\_test\_data.py，执行过程中会打印出来一些调试信息。



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64 | *#coding=utf8*    **import** sys    *#home\_dir = "D:/source/NLP/people\_daily//"*    home\_dir = "./"    **def** saveDataFile(trainobj,testobj,isTest,word,handle):  **if** isTest:          saveTrainFile(testobj,word,handle)  **else**:          saveTrainFile(trainobj,word,handle)    **def** saveTrainFile(fiobj,word,handle):  **if** len(word) > 0 **and**  word != "。" **and** word != "，":          fiobj.write(word + '\t' + handle  + '\n')  **else**:          fiobj.write('\n')    **def** convertTag():      fiobj    = open( home\_dir + 'people-daily.txt','r')      trainobj = open( home\_dir +'train.data','w' )      testobj  = open( home\_dir  +'test.data','w')        arr = fiobj.readlines()      i = 0  **for** a **in** sys.stdin:          i += 1          a = a.strip('\r\n\t ')  **if** a=="":**continue**          words = a.split(" ")          test = **False**  **if** i % 10 == 0:              test = **True**  **for** word **in** words[1:]:              print "---->", word              word = word.strip('\t ')  **if** len(word) > 0:                  i1 = word.find('[')  **if** i1 >= 0:                  word = word[i1+1:]              i2 = word.find(']')  **if** i2 > 0:                  w = word[:i2]              word\_hand = word.split('/')              print "----",word              w,h = word\_hand  *#print w,h*  **if** h == 'nr':    *#ren min*  *#print 'NR',w*  **if** w.find('·') >= 0:                      tmpArr = w.split('·')  **for** tmp **in** tmpArr:                          saveDataFile(trainobj,testobj,test,tmp,h)  **continue**              saveDataFile(trainobj,testobj,test,w,h)          saveDataFile(trainobj, testobj, test,"","")        trainobj.flush()      testobj.flush()    **if** \_\_name\_\_ == '\_\_main\_\_':      convertTag() |

## **执行训练和测试**

设置模板为：



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # Unigram  U00:*%x[-2,0]*  U01:*%x[-1,0]*  U02:*%x[0,0]*  U03:*%x[1,0]*  U04:*%x[2,0]*  U05:*%x[-1,0]/%x[0,0]*  U06:*%x[0,0]/%x[1,0]* |

训练的时候的-p参数根据自己机器情况设置



|  |  |
| --- | --- |
| 1  2 | crf\_learn -f 3 -p 4 -c 4.0 template train.data model > train.rst  crf\_test -m model **test**.data > **test**.rst |

## **计算准确率**

通过命令：python clc\_f.py  test.rst 执行python脚本，clc\_f.py中的具体程序：



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | *#!/usr/bin/python*  *# -\*- coding: utf-8 -\*-*  **import** sys  **if** \_\_name\_\_=="\_\_main\_\_":  **try**:          file = open(sys.argv[1], "r")  **except**:          print "result file is not specified, or open failed!"          sys.exit()      wc = 0      wc\_of\_test = 0      wc\_of\_gold = 0      wc\_of\_correct = 0      flag = **True**    **for** l **in** file:  **if** l=='\n': **continue**            \_, g, r = l.strip().split()    **if** r != g:              flag = **False**     wc += 1  **if** flag:              wc\_of\_correct +=1          flag = **True**        print "WordCount from result:", wc      print "WordCount of correct post :", wc\_of\_correct    *#准确率*      P = wc\_of\_correct/float(wc)        print "准确率:%f" % (P) |

## **实验结果**



|  |  |
| --- | --- |
| 1  2  3 | WordCount from result: 98369  WordCount of correct post : 91865  准确率:0.933882 |