使用CRF++训练分词模型

0. 训练前准备

- python2
- CRF++安装：https://taku910.github.io/crfpp/#download
- 训练语料，如 http://sighan.cs.uchicago.edu/bakeoff2005/ 和 http://www.icl.pku.edu.cn/icl_res/

1. 准备训练数据

假设现在已有标记好的分词训练语料，格式为每行都是一个句子，分好的词之间用空格分割。每一行的格式如格式一所示：

格式一：

```
共同 创造 美好 的 新 世纪 —— 二○○一年 新年 贺词
```

CRF++要求训练数据的格式是：每行都是 `字\t特征\t分词标记` 这种形式，中间的特征可以有很多列，最后一列是人工标记的分词结果，行与行之间用空行分割。如格式二所示：

格式二：各列分别为 `字|字的属性|前向最大匹配tag|后向最大匹配tag|人工标记结果tag`

```
  （      PUNC    S       S       S
附      CN      B       S       S
图      CN      E       B       B
片      CN      S       E       E
```

ps：这里使用4tag的形式，即B，M，E，S分别表示词首，词中，词尾，和单个字的词。

运行脚本：

```
python make_crf_train_data_multi.py train_file crf_train_file
```

即可得到符合要求的训练数据。其中

`train_file` 为符合上述格式一的训练语料，

`crf_train_file` 为符合上述格式二的可用于CRF++训练的数据格式。

测试数据要求跟训练数据格式一致，也可使用这个脚本进行转换。

2. 利用CRF++训练得到分词模型

使用crf_learn命令训练：

```
crf_learn -f 3 -c 4.0 -m 100 -t template crf_train_file crf-cws-model
```

其中：

`-m` 表示训练时的最大迭代次数

`-t` 表示生成文本形式的模型文件

各参数详细意义可以通过 `crf_learn -h` 进行查看。

`template` 特征模版文件需要预先定义好，这个文件决定了在训练和测试的时候使用哪些特征。

关于特征模版文件的解释可参考：https://taku910.github.io/crfpp/#templ

3. 对模型进行测试

使用 `crf_test` 命令来测试模型的效果。

```
crf_test -m crf-cws-model crf-test-file > crf-cws-res.utf8
```

`crf-test-file` 为训练数据，格式跟测试数据一致。

`crf-cws-res.utf8` 为利用前面训练得到的模型进行分词的结果，其在 `crf-test-file` 文件每一行的最后加了一列，即利用模型分词得到的结果的tag。

对分词结果进行评判

利用脚本 `cws_res_calculate.py` .

```
python cws_res_calculate.py crf-cws-res.utf8
```

4. 使用模型进行分词

要使用前面训练得到的模型来进行分词，可以利用脚本 `crf_cws.py` 。

`python crf_cws.py crf-cws-model crf_test.utf8 crf_test_res.utf8` ,其中

`crf-cws-model` 为前面训练得到的分词模型，

`crf_test.utf8` 为需要分词的文本，每行为一句话。

`crf_test_res.utf8` 为模型分词的结果。