**SpaCy中文模型训练工作流程**

**获得预处理的中文维基百科语料库**

有关更多详细信息，请参见project [chinese-wikipedia-corpus-creator](https://github.com/howl-anderson/chinese-wikipedia-corpus-creator)。

**产生维基百科语料库**

* 输入：-
* 输出： token\_cleaned\_plain\_files/
* 脚本： create\_wikipedia\_corpus.bash

**将语料库复制到工作区**

* 输入：`chinese-wikipedia-corpus-creator / token\_cleaned\_plain\_files /``
* 输出： token\_cleaned\_plain\_files/
* 脚本： move\_wikipedia\_corpus.bash

**计算字频**

* 输入： token\_cleaned\_plain\_files/\*
* 输出： WORDS\_FREQ.txt
* 脚本： compute\_words\_freq.bash

**将所有文件合并为一个**

* 输入： token\_cleaned\_plain\_files/\*
* 输出： WORDS.txt
* 脚本： merge\_all\_text\_files.bash

**计算棕色簇**

**棕色集群计算软件**

官方软件是[browncluster](https://github.com/percyliang/brown-cluster)。

**安装**

* 输入：-
* 输出：
* 脚本： download\_and\_compile\_brown\_cluster.bash

**计算**

* 输入： WORDS.txt
* 输出： WORDS-c1000-p1.out/\*
* 脚本： compute\_brown\_cluster.bash

**计算词向量**

* 输入： token\_cleaned\_plain\_files/\*
* 输出： WORDS\_VECS.txt
* 脚本： compute\_plain\_word\_vec.bash

**初始SpaCy模型**

**建立基础模型**

* 输入： ./WORDS-c1000-p1.out/paths WORDS\_VECS.txt WORDS\_FREQ.txt
* 输出： spacy\_models/base\_model/\*\*/\*
* 脚本： create\_init\_model.bash

**修改型号名称**

* 输入： spacy\_models/base\_model/meta.json
* 输出： spacy\_models/base\_model/meta.json
* 脚本： update\_model\_meta.py

**获取UD\_Chinese-GSD语料库**

**下载**

* 输入：-
* 输出： corpus/UD\_Chinese-GS.zip
* 脚本： download\_UD\_Chinese-GSD\_corpus.bash

**提取**

* 输入： corpus/UD\_Chinese-GSd.zip
* 输出： corpus/UD\_Chinese-GSd
* 脚本： extract\_UD\_Chinese-GSD\_corpus.bash

**转换成简体中文**

* 输入： corpus/UD\_Chinese-GSd/zh-ud-\*.conllu
* 输出： corpus/UD\_Chinese-GSd/zh-simplified-ud-\*.conllu
* 脚本： convert\_UD\_Chinese-GSD\_corpus.bash

**转换UD语料库格式**

* 输入： .corpus/UD\_Chinese-GSD/zh-simplified-ud-\*.conllu
* 输出： corpus/spacy/zh-simplified-ud-\*.conllu
* 脚本： format\_convertor.bash

**带有单词向量，单词簇和单词频率的初始spacy模型**

* 输入：WORDS\_FREQ.txt，WORDS-c1000-p1.out/paths，WORDS\_VECS.txt
* 输出： zh\_model/\*
* 脚本： init\_model.bash

**为POS和依赖解析器训练SpaCy模型**

* 输入： zh\_model corpus/spacy/zh-simplified-ud-\*.conllu
* 输出： dependency\_model
* 脚本： train\_model.bash

**将onotNote 5转换为spacy json文件**

* 输入： TODO
* 输出： TODO
* 脚本： onto\_to\_spacy\_json.bash

**为NER解析器训练SpaCy模型**

* 输入： zh\_model china\_ner\_train.json china\_ner\_eval.json
* 输出： ner\_model
* 脚本： train\_ner.bash

**合并子模型**

* 输入：spacy\_models/dependency\_model，spacy\_models/ner\_model
* 输出： spacy\_models/final\_model
* 脚本： merge\_submodel.py

**创建包装**

* 输入： spacy\_models/final\_model/
* 输出： spacy\_package/
* 脚本： ./create\_model\_package.bash