实验题目

- 不同词性标注方法的实施和对比
- 编程实践 `CRF`

实验内容

- 1.利用 `Chinese.txt` 和 `English.txt` 的中英文句子，在实验二的基础上，继续利用以下给定的中英文工具进行词性标注。并对不同工具产生的结果进行简要对比分析，将实验过程与结果写成实验报告，实验课结束后提交
- 2.使用 `python` 编程实践 `CRF` ，进行词性标注。该实验基于 `python3.6` 以及 `keras` 训练 `bi-lstm` ,结合 `CRF` 来实现词性标注，本实验可以在云平台进行，也可以在在本机进行。

遇到和解决的问题

问题一

- 问题：不同词性标注方法采用不同的词性表，它们表示词性的方式都不相同，看不懂输出的结果
- 解决：对每种标注方法，搜索 `方法名 + 词性表` 基本都能得到其对应的词性表，也就是词性标注结果的解释，这样才能分析实验结果。

问题二

- 问题：CRF环境配置问题
- 解决方法如下所示

按实验指导书执行一遍

- `pip install keras-preprocessing==1.0.9` 。这个版本的貌似找不到了，直接 `pip install keras-preprocessing` 就好了。
- `unzip work/keras-contrib-master.zip` 。这个 `work/` 是工作目录下的意思，直接 `cd word` 进入存储文件的目录解压即可
- `python keras-contrib-master/setup.py install` 。这个就是你刚刚解压的 `zip` 生成的文件夹 `keras-contrib-master` ，直接执行命令即可
- `unzip work/bi-lstm-crf-master.zip`
- `python bi-lstm-crf-master/setup.py install` 。这两步和刚刚的两步是一个道理。

额外步骤

更换 keras 版本

- 运行 `python keras-contrib-master/setup.py install` 会安装上最新版本的 `keras==2.8.0`
- 没有安装 `tensorflow` ，如果安装上 `keras==2.8.0` 对应版本的 `tensorflow` 是无法跑通代码的
- 于是用了比较常用的 `tesorflow==1.14 + keras==2.2.5` ，可以跑通。

注意

`path1和path2` 需要改成自己相应文件的对应路径，比如我的是 `path1 = "/home/nsy/nlp/exp3/CRF/bi-lstm-crf-master"`

问题三

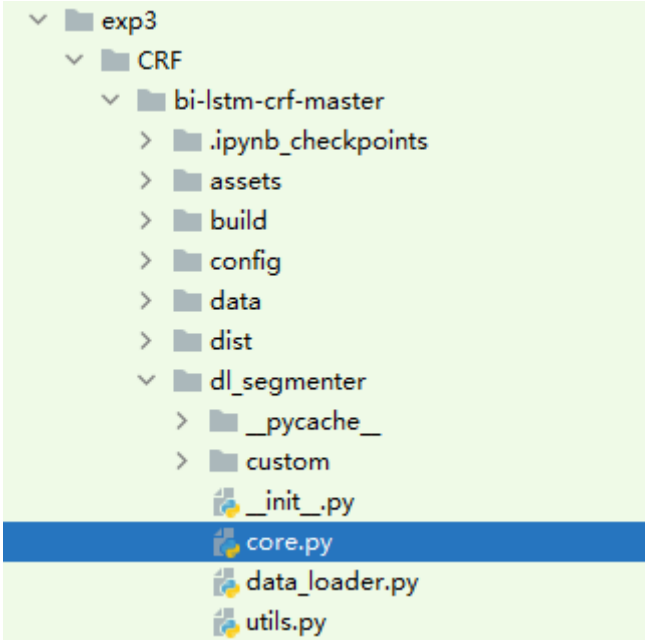
- 问题：在训练和测试过程中， `get_or_create` 函数均找不到预训练模型
- 解决方法如下

  在使用 `get_or_create` 加在模型时，总会报错 `no weights found,create a new model` 。这在训练和测试时都会出现

```
15  segmenter: DLSegmenter = get_or_create("bi-lstm-crf-master/config/default-config.json",
16                                         src_dict_path="bi-lstm-crf-master/config/src_dict.json",
17                                         tgt_dict_path="bi-lstm-crf-master/config/tgt_dict.json",
18                                         weights_path="bi-lstm-crf-master/models/weights.32--0.18.h5")
```

  但是，路径和环境都是没有问题的，于是去找源码，看看 `get_or_create` 函数是怎么定义的

  - 在 `CRF/bi-lstm-crf-master/dl_segmenter/core.py` 中找到相关内容




```
53      if weights_path is not None:
54          try:
55              self.model.load_weights(weights_path)
56              logging.info("weights loaded!")
57          except:
58              logging.error("No weights found, create a new model.")
```

我们发现错误是从这里来的，`try`没有成功，但是我们现在想看看`try`为什么不成功，而且在测试时，我们是一定要加载预训练模型的。于是改成

```
53      if weights_path is not None:
54          # try:
55          self.model.load_weights(weights_path)
56          logging.info("weights loaded!")
57          # except:
58          #     logging.error("No weights found, create a new model.")
```

再次运行测试代码`test.py`，发现报错

```
Traceback (most recent call last):
  File "/home/nsy/nlp/exp3/CRF/bi-lstm-crf-
master/dl_segmenter/core.py", line 188, in get_or_create
    DLSegmenter.__singleton = DLSegmenter(**config)
  File "/home/nsy/nlp/exp3/CRF/bi-lstm-crf-
master/dl_segmenter/core.py", line 55, in __init__
    self.model.load_weights(weights_path)
  File "/home/nsy/anaconda3/envs/nlp/lib/python3.6/site-
packages/keras/engine/saving.py", line 458, in load_wrapper
    return load_function(*args, **kwargs)
  File "/home/nsy/anaconda3/envs/nlp/lib/python3.6/site-
packages/keras/engine/network.py", line 1217, in load_weights
    f, self.layers, reshape=reshape)
  File "/home/nsy/anaconda3/envs/nlp/lib/python3.6/site-
packages/keras/engine/saving.py", line 1145, in
load_weights_from_hdf5_group
    original_keras_version =
f.attrs['keras_version'].decode('utf8')
AttributeError: 'str' object has no attribute 'decode'
```

我们定位到错误位置 `keras/engine/saving.py`

```
1144    if 'keras_version' in f.attrs:
1145        original_keras_version = f.attrs['keras_version'].decode('utf8')
1146    else:
1147        original_keras_version = '1'
1148    if 'backend' in f.attrs:
1149        original_backend = f.attrs['backend'].decode('utf8')
1150    else:
1151        original_backend = None
```

这是 `Python2和Python3字符编码上的区别`

- `Python3` 的 `str` 默认不是 `bytes`，所以不能 `decode`，只能先 `encode` 转为 `bytes`，再 `decode`
  `python2` 的 `str` 默认是 `bytes`，所以能 `decode`

因此，我们将原代码改为

```
1144    if 'keras_version' in f.attrs:
1145        original_keras_version = f.attrs['keras_version'].encode('utf8').decode('utf8')
1146    else:
1147        original_keras_version = '1'
1148    if 'backend' in f.attrs:
1149        original_backend = f.attrs['backend'].encode('utf8').decode('utf8')
1150    else:
1151        original_backend = None
```

再次运行测试文件 `test.py`，发现运行成功，不再报错。至此，我们解决了不能加载模型的问题。

问题四

- 问题：运行代码时显示找不到`module dl_segmenter`
- 解决：`dl_segmenter`是自定义的库，在`/home/nsy/nlp/exp3/CRF/bi-lstm-crf-master`下，因此将`/home/nsy/nlp/exp3/CRF/bi-lstm-crf-master`添加到`sys.path`中即可解决

实验步骤

英文词性标注

NLTK

- 词性表

Number	Tag	Description	中文翻译
1.	CC	Coordinating conjunction	对等连词，对等连词
2.	CD	Cardinal number	基数
3.	DT	Determiner	限定词（置于名词前起限定作用，如 the、some、my 等）
4.	EX	Existential there	存在句
5.	FW	Foreign word	外来语；外来词；外文原词
6.	IN	Preposition or subordinating conjunction	介词或者从属连词
7.	JJ	Adjective	形容词
8.	JJR	Adjective, comparative	比较级形容词，例子：better（更好的）
9.	JJS	Adjective, superlative	最高级形容词，例子：best（最好的）
10.	LS	List item marker	列表项标记
11.	MD	Modal	情态动词，在语法中，情态动词或情态助动词是一个如“can”或“would”之类的词，它与主动词连用，用来表达可能性、意图或必要性等观点
12.	NN	Noun, singular or mass	名词，单数或质量
13.	NNS	Noun, plural	名词,复数
14.	NNP	Proper noun, singular	专有名词,单数
15.	NNPS	Proper noun, plural	专有名词,复数

Number	Tag	Description	中文翻译
16.	PDT	Predeterminer	前位限定词，在语法中，一个前位限定词是一个词，使用在限定词之前，但仍然是名词组的一部分。例如，‘all’ in ‘all the time’ and ‘both’ in 'both our children’是前位限定词。
17.	POS	Possessive ending	所有格结尾。全部写法：所有格的写法一般是用“'s”和“of”表示。一般写法：一般常用：“'s”表示
18.	PRP	Personal pronoun	人称代词
19.	PRP$	Possessive pronoun	物主代词，物主代词有形容词性（my, your等）和名词性（mine, yours等）两种，形容词性的物主代词属于限定词。
20.	RB	Adverb	副词
21.	RBR	Adverb, comparative	副词、比较级
22.	RBS	Adverb, superlative	副词,最高级
23.	RP	Particle	小品词（与动词构成短语动词的副词或介词）
24.	SYM	Symbol	符号
25.	TO	to	到
26.	UH	Interjection	感叹词；感叹语
27.	VB	Verb, base form	动词原形
28.	VBD	Verb, past tense	动词过去式
29.	VBG	Verb, gerund or present participle	动词、动名词或现在分词
30.	VBN	Verb, past participle	动词过去分词

Number	Tag	Description	中文翻译
31.	VBP	Verb, non-3rd person singular present	动词，非第三人称单数现在时
32.	VBZ	Verb, 3rd person singular present	动词，第三人称单数现在时
33.	WDT	Wh-determiner	WH限定词
34.	WP	Wh-pronoun	WH-代词
35.	WP$	Possessive wh-pronoun	所有格wh-代词
36.	WRB	Wh-adverb	WWh-副词

分不出标点

- 代码

```
def nltk_nlp(data):
    ans = nltk.word_tokenize(data)
    tagged = nltk.pos_tag(ans)
    print(tagged)
```

- 结果
  - 优点：区分出了动词过去式和动词过去分词，对专有名词的识别效果不错，对动词的不同形式区分比较细致。
  - 缺点：`'Xi'` 和 `'Jinping'` 词性不一致。容易将名次和动词分错，比如 `('Xi', 'VB')` 就是将名词标成了动词原形。将太多的词标注成专有名词，不能进一步区分不同的专有名词，而且容易将首字母大写的词标注成专有名词。不能对标点符号进行标注。

```
[('Xi', 'NN'), ('Jinping', 'NNP'), (',', ','), ('male', 'NN'),
(',', ','), ('Han', 'NNP'), ('ethnicity', 'NN'), (',', ','),
('was', 'VBD'), ('born', 'VBN'), ('in', 'IN'), ('June', 'NNP'),
('1953', 'CD'), ('and', 'CC'), ('is', 'VBZ'), ('from', 'IN'),
('Fuping', 'VBG'), (',', ','), ('Shaanxi', 'NNP'), ('Province',
'NNP'), ('.', '.'), ('He', 'PRP'), ('began', 'VBD'), ('his',
'PRP$'), ('first', 'JJ'), ('job', 'NN'), ('in', 'IN'), ('January',
'NNP'), ('1969', 'CD'), ('and', 'CC'), ('joined', 'VBD'), ('the',
'DT'), ('Communist', 'NNP'), ('Party', 'NNP'), ('of', 'IN'),
('China', 'NNP'), ('(', '('), ('CPC', 'NNP'), (')', ')'), ('in',
'IN'), ('January', 'NNP'), ('1974', 'CD'), ('.', '.'), ('Xi',
'VB'), ('graduated', 'VBN'), ('from', 'IN'), ('School', 'NNP'),
('of', 'IN'), ('Humanities', 'NNP'), ('and', 'CC'), ('Social',
'NNP'), ('Sciences', 'NNPS'), (',', ','), ('Tsinghua', 'NNP'),
('University', 'NNP'), ('where', 'WRB'), ('he', 'PRP'),
('completed', 'VBD'), ('an', 'DT'), ('in-service', 'JJ'),
('graduate', 'NN'), ('program', 'NN'), ('in', 'IN'), ('Marxist',
'NNP'), ('theory', 'NN'), ('and', 'CC'), ('ideological', 'JJ'),
('and', 'CC'), ('political', 'JJ'), ('education', 'NN'), ('.',
'.'), ('He', 'PRP'), ('holds', 'VBZ'), ('a', 'DT'), ('Doctor',
'NNP'), ('of', 'IN'), ('Law', 'NNP'), ('degree', 'NN'), ('.',
'.'), ('Xi', 'NN'), ('is', 'VBZ'), ('currently', 'RB'),
('General', 'NNP'), ('Secretary', 'NNP'), ('of', 'IN'), ('the',
'DT'), ('CPC', 'NNP'), ('Central', 'NNP'), ('Committee', 'NNP'),
(',', ','), ('Chairman', 'NNP'), ('of', 'IN'), ('the', 'DT'),
('CPC', 'NNP'), ('Central', 'NNP'), ('Military', 'NNP'),
('Commission', 'NNP'), (',', ','), ('President', 'NNP'), ('of',
'IN'), ('the', 'DT'), ('People', 'NNP'), ("'s", 'POS'),
('Republic', 'NNP'), ('of', 'IN'), ('China', 'NNP'), ('(', '('),
('PRC', 'NNP'), (')', ')'), (',', ','), ('and', 'CC'),
('Chairman', 'NNP'), ('of', 'IN'), ('the', 'DT'), ('PRC', 'NNP'),
('Central', 'NNP'), ('Military', 'NNP'), ('Commission', 'NNP'),
('.', '.')]
```

Spacy

- 词性表
  - `ADJ`：形容词
  - `ADP`：adposition
  - `ADV`：副词
  - `AUX`：辅助动词
  - `CONJ`：协调会议
  - `DET`：determininer
  - `INTJ`：感叹词
  - `NOUN`：名词
  - `NUM`：数字

    - `PART`：particle
    - `PRON`：代词
    - `PROPN`：专有名词
    - `PUNCT`：标点符号
    - `SCONJ`：从属连接
    - `SYM`：symbol
    - `VERB`：动词
    - `X`：其他
    - 代码

- ```
  def spacy_nlp(data):
      nlp = spacy.load("en_core_web_sm")
      doc = nlp(data)
      ans = [token.text for token in doc]
      tagged = [{token.text: token.pos_} for token in doc]
      print(tagged)
  ```

- 结果
    - 优点：能够对标点符号进行标注。能够区分名词和专有名词，比如将 `Xi Jinping` 标注成了专有名词而不是名词
    - 缺点：将太多的词标注为 `'PROPN'`,而不能将其中的词进一步细分
  ```

```
[{'Xi': 'PROPN'}, {'Jinping': 'PROPN'}, {',': 'PUNCT'}, {'male':
'NOUN'}, {',': 'PUNCT'}, {'Han': 'PROPN'}, {'ethnicity': 'NOUN'},
{',': 'PUNCT'}, {'was': 'AUX'}, {'born': 'VERB'}, {'in': 'ADP'},
{'June': 'PROPN'}, {'1953': 'NUM'}, {'and': 'CCONJ'}, {'is':
'AUX'}, {'from': 'ADP'}, {'Fuping': 'PROPN'}, {',': 'PUNCT'},
{'Shaanxi': 'PROPN'}, {'Province': 'PROPN'}, {'.': 'PUNCT'},
{'He': 'PRON'}, {'began': 'VERB'}, {'his': 'DET'}, {'first':
'ADJ'}, {'job': 'NOUN'}, {'in': 'ADP'}, {'January': 'PROPN'},
{'1969': 'NUM'}, {'and': 'CCONJ'}, {'joined': 'VERB'}, {'the':
'DET'}, {'Communist': 'PROPN'}, {'Party': 'PROPN'}, {'of': 'ADP'},
{'China': 'PROPN'}, {'(': 'PUNCT'}, {'CPC': 'PROPN'}, {')':
'PUNCT'}, {'in': 'ADP'}, {'January': 'PROPN'}, {'1974': 'NUM'},
{'.': 'PUNCT'}, {'Xi': 'PROPN'}, {'graduated': 'VERB'}, {'from':
'ADP'}, {'School': 'PROPN'}, {'of': 'ADP'}, {'Humanities':
'PROPN'}, {'and': 'CCONJ'}, {'Social': 'PROPN'}, {'Sciences':
'PROPN'}, {',': 'PUNCT'}, {'Tsinghua': 'PROPN'}, {'University':
'PROPN'}, {'where': 'ADV'}, {'he': 'PRON'}, {'completed': 'VERB'},
{'an': 'DET'}, {'in': 'ADP'}, {'-': 'PUNCT'}, {'service': 'NOUN'},
{'graduate': 'NOUN'}, {'program': 'NOUN'}, {'in': 'ADP'},
{'Marxist': 'ADJ'}, {'theory': 'NOUN'}, {'and': 'CCONJ'},
{'ideological': 'ADJ'}, {'and': 'CCONJ'}, {'political': 'ADJ'},
{'education': 'NOUN'}, {'.': 'PUNCT'}, {'He': 'PRON'}, {'holds':
'VERB'}, {'a': 'DET'}, {'Doctor': 'PROPN'}, {'of': 'ADP'}, {'Law':
'PROPN'}, {'degree': 'NOUN'}, {'.': 'PUNCT'}, {'Xi': 'PROPN'},
{'is': 'AUX'}, {'currently': 'ADV'}, {'General': 'PROPN'},
{'Secretary': 'PROPN'}, {'of': 'ADP'}, {'the': 'DET'}, {'CPC':
'PROPN'}, {'Central': 'PROPN'}, {'Committee': 'PROPN'}, {',':
'PUNCT'}, {'Chairman': 'PROPN'}, {'of': 'ADP'}, {'the': 'DET'},
{'CPC': 'PROPN'}, {'Central': 'PROPN'}, {'Military': 'PROPN'},
{'Commission': 'PROPN'}, {',': 'PUNCT'}, {'President': 'PROPN'},
{'of': 'ADP'}, {'the': 'DET'}, {'People': 'PROPN'}, {"'s":
'PART'}, {'Republic': 'PROPN'}, {'of': 'ADP'}, {'China': 'PROPN'},
{'(': 'PUNCT'}, {'PRC': 'PROPN'}, {')': 'PUNCT'}, {',': 'PUNCT'},
{'and': 'CCONJ'}, {'Chairman': 'PROPN'}, {'of': 'ADP'}, {'the':
'DET'}, {'PRC': 'PROPN'}, {'Central': 'PROPN'}, {'Military':
'PROPN'}, {'Commission': 'PROPN'}, {'.': 'PUNCT'}]
```

StanfordCoreNLP

- 词性表和 `NLTK` 相同
- 代码

```
def stanford_nlp(data):
    # _*_coding:utf-8_*_
    with StanfordCoreNLP(r'D:\stanford-corenlp-full-2018-02-27')
as nlp:
        print(nlp.pos_tag(data))
```

- 结果
  - 优点：相较于 `NLTK` 来说，将 `Xi` 和 `Jinping` 标注成一个词性是进步的。将 `Fuping` 标注成 `NNP` 而不是 `VBG` 也是进步的，没有出现 `NLTK` 中 `('Xi','VB')` 的情况。整体效果相较于 `NLTK` 来说都是有长足进步的。
  - 缺点：和 `NLTK` 一样，对名词的区分度不强，专有名词不能再细分，名词与专有名词容易搞混

```
[('Xi', 'NN'), ('Jinping', 'NN'), (',', ','), ('male', 'NN'),
(',', ','), ('Han', 'NNP'), ('ethnicity', 'NN'), (',', ','),
('was', 'VBD'), ('born', 'VBN'), ('in', 'IN'), ('June', 'NNP'),
('1953', 'CD'), ('and', 'CC'), ('is', 'VBZ'), ('from', 'IN'),
('Fuping', 'NNP'), (',', ','), ('Shaanxi', 'NNP'), ('Province',
'NNP'), ('.', '.'), ('He', 'PRP'), ('began', 'VBD'), ('his',
'PRP$'), ('first', 'JJ'), ('job', 'NN'), ('in', 'IN'), ('January',
'NNP'), ('1969', 'CD'), ('and', 'CC'), ('joined', 'VBD'), ('the',
'DT'), ('Communist', 'NNP'), ('Party', 'NNP'), ('of', 'IN'),
('China', 'NNP'), ('(', '-LRB-'), ('CPC', 'NNP'), (')', '-RRB-'),
('in', 'IN'), ('January', 'NNP'), ('1974', 'CD'), ('.', '.'),
('Xi', 'NN'), ('graduated', 'VBD'), ('from', 'IN'), ('School',
'NNP'), ('of', 'IN'), ('Humanities', 'NNPS'), ('and', 'CC'),
('Social', 'NNP'), ('Sciences', 'NNPS'), (',', ','), ('Tsinghua',
'NNP'), ('University', 'NNP'), ('where', 'WRB'), ('he', 'PRP'),
('completed', 'VBD'), ('an', 'DT'), ('in-service', 'JJ'),
('graduate', 'NN'), ('program', 'NN'), ('in', 'IN'), ('Marxist',
'JJ'), ('theory', 'NN'), ('and', 'CC'), ('ideological', 'JJ'),
('and', 'CC'), ('political', 'JJ'), ('education', 'NN'), ('.',
'.'), ('He', 'PRP'), ('holds', 'VBZ'), ('a', 'DT'), ('Doctor',
'NN'), ('of', 'IN'), ('Law', 'NN'), ('degree', 'NN'), ('.', '.'),
('Xi', 'NN'), ('is', 'VBZ'), ('currently', 'RB'), ('General',
'NNP'), ('Secretary', 'NNP'), ('of', 'IN'), ('the', 'DT'), ('CPC',
'NNP'), ('Central', 'NNP'), ('Committee', 'NNP'), (',', ','),
('Chairman', 'NNP'), ('of', 'IN'), ('the', 'DT'), ('CPC', 'NNP'),
('Central', 'NNP'), ('Military', 'NNP'), ('Commission', 'NNP'),
(',', ','), ('President', 'NNP'), ('of', 'IN'), ('the', 'DT'),
('People', 'NNS'), ("'s", 'POS'), ('Republic', 'NN'), ('of',
'IN'), ('China', 'NNP'), ('(', '-LRB-'), ('PRC', 'NNP'), (')', '-
RRB-'), (',', ','), ('and', 'CC'), ('Chairman', 'NNP'), ('of',
'IN'), ('the', 'DT'), ('PRC', 'NNP'), ('Central', 'NNP'),
('Military', 'NNP'), ('Commission', 'NNP'), ('.', '.')]
```

中文词性标注

jieba

- 词性表

标签	含义	标签	含义	标签	含义	标签	含义
n	普通名词	f	方位名词	s	处所名词	t	时间
nr	人名	ns	地名	nt	机构名	nw	作品名
nz	其他专名	v	普通动词	vd	动副词	vn	名动词
a	形容词	ad	副形词	an	名形词	d	副词
m	数量词	q	量词	r	代词	p	介词
c	连词	u	助词	xc	其他虚词	w	标点符号
PER	人名	LOC	地名	ORG	机构名	TIME	时间

- 代码(paddle)

```
def jieba_test(data):
    jieba.enable_paddle()
    words = pseg.cut(data, use_paddle=True)
    # words = pseg.cut(data)
    ans = []
    for word, flag in words:
        ans.append({word: flag})
    print(ans)
```

- 结果
  - 优点：对专有名词进行了细分，比如出现`TIME`，`LOC`，`ORG`，`PER`，分别是时间，地点，机构名，人名，而不是将这些统称为专有名词
  - 缺点：对标点的标注比较混乱，有时候标注成名词，有时候标注成动词。有时分词不太准确，搞不清动词和动名词。

```
[{'3月23日下午': 'TIME'}, {'，“': 'v'}, {'青年': 'n'}, {'报杯赛':
'n'}, {'”U': 'v'}, {'19': 'm'}, {'邀请赛': 'n'}, {'在': 'p'}, {'越
南': 'LOC'}, {'芽庄': 'LOC'}, {'进行': 'v'}, {'，': 'n'}, {'前':
'f'}, {'国脚': 'n'}, {'曲波': 'PER'}, {'挂帅': 'v'}, {'的': 'u'},
{'中国': 'LOC'}, {'U': 'v'}, {'19': 'm'}, {'B队': 'n'}, {'迎战':
'v'}, {'泰国': 'LOC'}, {'U': 'v'}, {'19': 'm'}, {'，': 'v'}, {'上半
场': 'TIME'}, {'国青队': 'ORG'}, {'的': 'u'}, {'门户': 'n'}, {'大
开，': 'v'}, {'泰国': 'LOC'}, {'在': 'p'}, {'第11分钟': 'TIME'},
{'和': 'c'}, {'第17分钟': 'TIME'}, {'连': 'd'}, {'进': 'v'}, {'2':
'm'}, {'球': 'n'}, {'，': 'v'}, {'半场': 'n'}, {'国青': 'n'}, {'0':
'm'}, {'射门': 'v'}, {'0': 'm'}, {'角球': 'n'}, {'，': 'v'}, {'几
乎': 'd'}, {'被': 'p'}, {'完全': 'd'}, {'压制': 'v'}, {'。': 'v'},
{'下半场': 'TIME'}, {'，国青': 'PER'}, {'的': 'u'}, {'进攻': 'vn'},
{'一度': 'd'}, {'有所': 'v'}, {'起色': 'n'}, {'，': 'n'}, {'并':
'c'}, {'由': 'p'}, {'马辅渔': 'PER'}, {'利用': 'v'}, {'远射': 'vn'},
{'扳回': 'v'}, {'一': 'm'}, {'球': 'n'}, {'，': 'v'}, {'但': 'c'},
{'最终': 'ad'}, {'未能': 'v'}, {'扳平': 'v'}, {'比分': 'n'}, {'。':
'v'}, {'全场': 'n'}, {'比赛': 'vn'}, {'结束': 'v'}, {'，国青1-2输
球，': 'nz'}, {'继': 'v'}, {'中国': 'LOC'}, {'杯': 'n'}, {'国足':
'ORG'}, {'0': 'm'}, {'-': 'w'}, {'1': 'm'}, {'输': 'v'}, {'给':
'p'}, {'泰国': 'LOC'}, {'之后': 'f'}, {'，': 'v'}, {'3天内':
'TIME'}, {'遭遇': 'v'}, {'泰国': 'LOC'}, {'足球': 'n'}, {'双杀':
'v'}, {'。': 'n'}]
```

- 代码(默认)

```python
def jieba_test(data):
    # jieba.enable_paddle()
    # words = pseg.cut(data, use_paddle=True)
    words = pseg.cut(data)
    ans = []
    for word, flag in words:
        ans.append({word: flag})
    print(ans)
```

- 结果
  - 分词十分混乱，效果明显不如`paddle`模式，这里不做评价

```
[{'3': 'm'}, {'月': 'm'}, {'23': 'm'}, {'日': 'm'}, {'下午': 't'},
{',': 'x'}, {'“': 'x'}, {'青年报': 'n'}, {'杯赛': 'n'}, {'”':
'x'}, {'U19': 'eng'}, {'邀请赛': 'n'}, {'在': 'p'}, {'越南': 'ns'},
{'芽庄': 'n'}, {'进行': 'v'}, {',': 'x'}, {'前': 'f'}, {'国脚':
'n'}, {'曲波': 'nr'}, {'挂帅': 'n'}, {'的': 'uj'}, {'中国': 'ns'},
{'U19B': 'eng'}, {'队': 'n'}, {'迎战': 'v'}, {'泰国': 'ns'},
{'U19': 'eng'}, {',': 'x'}, {'上半场': 'ns'}, {'国青队': 'nt'},
{'的': 'uj'}, {'门户': 'n'}, {'大': 'a'}, {'开': 'v'}, {',': 'x'},
{'泰国': 'ns'}, {'在': 'p'}, {'第': 'm'}, {'11': 'm'}, {'分钟':
'q'}, {'和': 'c'}, {'第': 'm'}, {'17': 'm'}, {'分钟': 'q'}, {'连
进': 'a'}, {'2': 'm'}, {'球': 'n'}, {',': 'x'}, {'半场': 'n'},
{'国青': 'nt'}, {'0': 'm'}, {'射门': 'n'}, {'0': 'x'}, {'角球':
'n'}, {',': 'x'}, {'几乎': 'd'}, {'被': 'p'}, {'完全': 'ad'}, {'压
制': 'n'}, {'。': 'x'}, {'下半场': 'n'}, {',': 'x'}, {'国青':
'nr'}, {'的': 'uj'}, {'进攻': 'v'}, {'一度': 'mq'}, {'有所': 'n'},
{'起色': 'n'}, {',': 'x'}, {'并': 'c'}, {'由': 'p'}, {'马辅渔':
'nr'}, {'利用': 'n'}, {'远射': 'v'}, {'扳回': 'v'}, {'一': 'm'},
{'球': 'n'}, {',': 'x'}, {'但': 'c'}, {'最终': 'd'}, {'未能': 'v'},
{'扳平': 'n'}, {'比分': 'd'}, {'。': 'x'}, {'全场': 'n'}, {'比赛':
'vn'}, {'结束': 'v'}, {',': 'x'}, {'国青': 'nt'}, {'1': 'm'}, {'-
': 'x'}, {'2': 'x'}, {'输球': 'n'}, {',': 'x'}, {'继': 'v'}, {'中
国': 'ns'}, {'杯国': 'n'}, {'足': 'a'}, {'0': 'm'}, {'-': 'x'},
{'1': 'x'}, {'输给': 'v'}, {'泰国': 'ns'}, {'之后': 'f'}, {',':
'x'}, {'3': 'm'}, {'天内': 's'}, {'遭遇': 'n'}, {'泰国': 'ns'}, {'足
球': 'n'}, {'双': 'm'}, {'杀': 'v'}, {'。': 'x'}]
```

StanfordCoreNLP

- 代码

```
def stanford_nlp(data):
    # _*_coding:utf-8_*_
    with StanfordCoreNLP(r'D:\stanford-corenlp-full-2018-02-27',
lang='zh') as nlp:
        print(nlp.pos_tag(data))
```

- 结果
  - 优点：对标点的词性划分比较正常
  - 缺点：对名次划分不够细致，很多词的词性都标注错误

```
[('3月', 'NT'), ('23日', 'NT'), ('下午', 'NT'), ('，', 'PU'), ('“', 'PU'), ('青年报', 'NN'), ('杯赛', 'NN'), ('”', 'PU'), ('U19', 'NN'), ('邀请赛', 'NN'), ('在', 'P'), ('越南', 'NR'), ('芽', 'NN'), ('庄', 'NR'), ('进行', 'VV'), ('，', 'PU'), ('前', 'JJ'), ('国脚', 'NN'), ('曲波', 'NR'), ('挂帅', 'VV'), ('的', 'DEC'), ('中国', 'NR'), ('U19', 'NN'), ('B', 'NN'), ('队', 'NN'), ('迎战', 'VV'), ('泰国', 'NR'), ('U19', 'NN'), ('，', 'PU'), ('上半场', 'NN'), ('国青队', 'NR'), ('的', 'DEG'), ('门户', 'NN'), ('大开', 'VV'), ('，', 'PU'), ('泰国', 'NR'), ('在', 'P'), ('第11', 'OD'), ('分钟', 'M'), ('和', 'CC'), ('第17', 'OD'), ('分钟', 'M'), ('连', 'AD'), ('进', 'VV'), ('2', 'CD'), ('球', 'NN'), ('，', 'PU'), ('半场', 'NN'), ('国青', 'NN'), ('0', 'CD'), ('射门', 'VV'), ('0', 'CD'), ('角球', 'NN'), ('，', 'PU'), ('几乎', 'AD'), ('被', 'LB'), ('完全', 'AD'), ('压制', 'VV'), ('。', 'PU'), ('下半场', 'NN'), ('，', 'PU'), ('国青', 'VA'), ('的', 'DEC'), ('进攻', 'NN'), ('一度', 'AD'), ('有所', 'VV'), ('起色', 'NN'), ('，', 'PU'), ('并', 'AD'), ('由', 'P'), ('马辅渔', 'NR'), ('利用', 'VV'), ('远射', 'NN'), ('扳回', 'VV'), ('一', 'CD'), ('球', 'NN'), ('，', 'PU'), ('但', 'AD'), ('最终', 'AD'), ('未', 'AD'), ('能', 'VV'), ('扳平', 'VV'), ('比分', 'NN'), ('。', 'PU'), ('全', 'DT'), ('场', 'NN'), ('比赛', 'NN'), ('结束', 'VV'), ('，', 'PU'), ('国青', 'NN'), ('1-2', 'CD'), ('输球', 'VV'), ('，', 'PU'), ('继', 'P'), ('中国', 'NR'), ('杯', 'NN'), ('国足', 'NN'), ('0-1', 'NN'), ('输给', 'VV'), ('泰国', 'NR'), ('之后', 'LC'), ('，', 'PU'), ('3', 'CD'), ('天', 'M'), ('内', 'LC'), ('遭遇', 'VV'), ('泰国', 'NR'), ('足球', 'NN'), ('双杀', 'NN'), ('。', 'PU')]
```

SnowNLP

- 代码

```python
def snow_nlp(data):
    s = SnowNLP(data)
    # print(s.words)
    ans = []
    for i in s.tags:
        ans.append(i)
    print(ans)
```

- 结果
  - 优点：总体完成了分词的任务，但是相较于其他方法没有明显的优点
  - 缺点：对名词的区分比较乱。由于分词效果不好，因此影响了词性标注的效果。

```
[('3', 'ns'), ('月', 'n'), ('23', 'Rg'), ('日', 'Ng'), ('下午',
't'), ('，“', 't'), ('青年报', 'n'), ('杯赛', 'Dg'), ('”U19',
'Ag'), ('邀请赛', 'n'), ('在', 'p'), ('越南', 'ns'), ('芽', 'n'),
('庄', 'nr'), ('进行', 'v'), ('，', 'w'), ('前', 'f'), ('国', 'n'),
('脚曲', 'nr'), ('波', 'nr'), ('挂帅', 'v'), ('的', 'u'), ('中国',
'ns'), ('U19B', 'nz'), ('队', 'n'), ('迎战', 'v'), ('泰国', 'ns'),
('U19，', 'nz'), ('上半场', 'n'), ('国', 'n'), ('青队', 'f'), ('的',
'u'), ('门户', 'n'), ('大', 'd'), ('开', 'v'), ('，', 'w'), ('泰国',
'ns'), ('在', 'p'), ('第', 'm'), ('11', 'm'), ('分钟', 'q'), ('和',
'c'), ('第', 'm'), ('17', 'm'), ('分钟', 'q'), ('连', 'n'), ('进',
'v'), ('2', 'Bg'), ('球', 'n'), ('，', 'w'), ('半场', 'nt'), ('国',
'n'), ('青', 'nr'), ('0', 'nr'), ('射门', 'v'), ('0', 'u'), ('角
球', 'n'), ('，', 'w'), ('几乎', 'd'), ('被', 'p'), ('完全', 'ad'),
('压制', 'v'), ('。', 'w'), ('下半场', 'n'), ('，', 'w'), ('国',
'n'), ('青', 'a'), ('的', 'u'), ('进攻', 'vn'), ('一度', 'd'), ('有
所', 'v'), ('起色', 'n'), ('，', 'w'), ('并', 'c'), ('由', 'p'),
('马', 'nr'), ('辅', 'nr'), ('渔', 'Ng'), ('利用', 'v'), ('远射',
'v'), ('扳回', 'v'), ('一', 'm'), ('球', 'n'), ('，', 'w'), ('但',
'c'), ('最终', 'd'), ('未', 'd'), ('能', 'v'), ('扳平', 'h'), ('比
分', 'n'), ('。', 'w'), ('全场', 'n'), ('比赛', 'vn'), ('结束',
'v'), ('，', 'w'), ('国', 'n'), ('青', 'nr'), ('1-2', 'nr'), ('输',
'v'), ('球', 'n'), ('，', 'w'), ('继', 'Vg'), ('中国', 'ns'),
('杯', 'q'), ('国', 'n'), ('足', 'd'), ('0-1', 'd'), ('输', 'v'),
('给', 'p'), ('泰国', 'ns'), ('之后', 'f'), ('，3', 'm'), ('天',
'q'), ('内', 'f'), ('遭遇', 'v'), ('泰国', 'ns'), ('足球', 'n'),
('双杀', 'Rg'), ('。', 'w')]
```

THULAC

- 词性表

符号	含义	符号	含义
n	名词	np	人名
mq	数量词	f	方位词
m	数词	q	量词
ni	机构名	r	代词
v	动词	a	形容词
w	标点	d	副词
p	介词	c	连词
u	助词	y	语气词
e	叹词	o	拟声词
i	成语	x	其他
j	简称	h	前接成分
k	后接成分	g	语素
s	处所词	w	标点符号
np	人名	ns	地名
ni	机构名称	nx	外文字符
nz	其它专名	h	前接成分

- 代码

```
def thulac_nlp(data):
    thu1 = thulac.thulac()  # 默认模式
    text = thu1.cut(data, text=True)  # 进行一句话分词
    print(text)
```

- 结果
  - 缺点：分词效果太细致，导致词性标注太细致，容易将原本是一个词性的词分开标注。部分词性标注不准确，将 `0` 标注为 `v` 。容易分不清副词和形容词。
  - 优点：不考虑分词的情况下，词性标注效果不错，可以区分人名，机构名，量词等。总体效果十分不错。

```
3月_t 23日_t 下午_t ， _w “_w 青年_n 报杯赛_n ”_w U_g 19_m 邀请赛_n 在_p 越南_ns 芽庄_ns 进行_v ， _w 前_f 国脚_n 曲波_np 挂帅_v 的_u 中国_ns U19B队_n 迎战_v 泰国_ns U19_x ， _w 上半场_n 国青队_ni 的_u 门户_n 大_d 开_v ， _w 泰国_ns 在_p 第11_m 分钟_q 和_c 第17_m 分钟_q 连_d 进_v 2_m 球_n ， _w 半_m 场_q 国青0_n 射门_v 0_v 角球_n ， _w 几乎_d 被_p 完全_a 压制_v 。_w 下半场_n ， _w 国青_a 的_u 进攻_v 一度_d 有所_v 起色_n ， _w 并_c 由_p 马辅渔_np 利用_v 远射_v 扳回_v 一_m 球_n ， _w 但_c 最终_d 未能_v 扳平_v 比分_n 。_w 全场_n 比赛_v 结束_v ， _w 国青_a 1_m -_w 2_m 输_v 球_n ， _w 继_g 中国_ns 杯国_ns 足0_m -_w 1_m 输_v 给_v 泰国_ns 之后_f ， _w 3_m 天_q 内_f 遭遇_v 泰国_ns 足球_n 双杀_a 。_w
```

NLPIR

- 代码

```
def pynlpir_nlp(data):
    pynlpir.open()
    ans = pynlpir.segment(data)
    print(ans)
```

- 结果
  - 优点：标注方式为正常的英文，而不是自定义的词性表，看起来非常易懂。
  - 缺点：没有对名词进行区分，和其他方法相比，甚至没有将名词和专有名词区分开。词性标注比较乱。

```
[('3月', 'time word'), ('23日', 'time word'), ('下午', 'time
word'), ('，', 'punctuation mark'), ('“', 'punctuation mark'),
('青年报', 'multiword expression'), ('杯', 'noun'), ('赛', 'verb'),
('”', 'punctuation mark'), ('U19', 'noun'), ('邀请赛', 'noun'),
('在', 'preposition'), ('越南', 'noun'), ('芽', 'noun'), ('庄',
'noun'), ('进行', 'verb'), ('，', 'punctuation mark'), ('前', 'noun
of locality'), ('国脚', 'noun'), ('曲波', 'noun'), ('挂帅',
'verb'), ('的', 'particle'), ('中国', 'noun'), ('U19B', 'noun'),
('队', 'noun'), ('迎战', 'verb'), ('泰国', 'noun'), ('U19',
'noun'), ('，', 'punctuation mark'), ('上半场', 'noun'), ('国青',
'noun'), ('队', 'noun'), ('的', 'particle'), ('门户', 'noun'),
('大', 'adverb'), ('开', 'verb'), ('，', 'punctuation mark'), ('泰
国', 'noun'), ('在', 'preposition'), ('第11', 'numeral'), ('分钟',
'classifier'), ('和', 'conjunction'), ('第17', 'numeral'), ('分钟',
'classifier'), ('连', 'verb'), ('进', 'verb'), ('2', 'numeral'),
('球', 'noun'), ('，', 'punctuation mark'), ('半', 'numeral'),
('场', 'classifier'), ('国青', 'noun'), ('0', 'numeral'), ('射门',
'verb'), ('0', 'numeral'), ('角球', 'noun'), ('，', 'punctuation
mark'), ('几乎', 'adverb'), ('被', 'preposition'), ('完全',
'adjective'), ('压制', 'verb'), ('。', 'punctuation mark'), ('下',
'verb'), ('半', 'numeral'), ('场', 'classifier'), ('，',
'punctuation mark'), ('国青', 'noun'), ('的', 'particle'), ('进攻',
'verb'), ('一度', 'adverb'), ('有所', 'verb'), ('起色', 'noun'),
('，', 'punctuation mark'), ('并', 'conjunction'), ('由',
'preposition'), ('马', 'noun'), ('辅', 'verb'), ('渔利', 'verb'),
('用', 'preposition'), ('远射', 'verb'), ('扳回', 'verb'), ('一',
'numeral'), ('球', 'noun'), ('，', 'punctuation mark'), ('但',
'conjunction'), ('最终', 'adverb'), ('未能', 'verb'), ('扳平',
'verb'), ('比分', 'noun'), ('。', 'punctuation mark'), ('全场',
'noun'), ('比赛', 'verb'), ('结束', 'verb'), ('，', 'punctuation
mark'), ('国青', 'noun'), ('1-2', 'numeral'), ('输', 'verb'),
('球', 'noun'), ('，', 'punctuation mark'), ('继', 'verb'), ('中
国', 'noun'), ('杯', 'noun'), ('国', 'noun'), ('足', 'adjective'),
('0-1', 'numeral'), ('输', 'verb'), ('给', 'preposition'), ('泰国',
'noun'), ('之后', 'noun of locality'), ('，', 'punctuation mark'),
('3', 'numeral'), ('天', 'classifier'), ('内', 'noun of
locality'), ('遭遇', 'verb'), ('泰国', 'noun'), ('足球', 'noun'),
('双', 'numeral'), ('杀', 'verb'), ('。', 'punctuation mark')]
```

CRF实践

实验过程大多是环境配置的过程，具体过程已经写在`遇到和解决的问题`中，实验环境的配置在此不再赘述。

预处理

- 配置实验环境
- 将标注的语料转化成BIS形式，合并在一个文件中
- 生成字典
- 转化成h5文件
- 配置相关参数

训练模型

- 加载数据
- 定义并训练模型

词性标注测试

- 测试模型

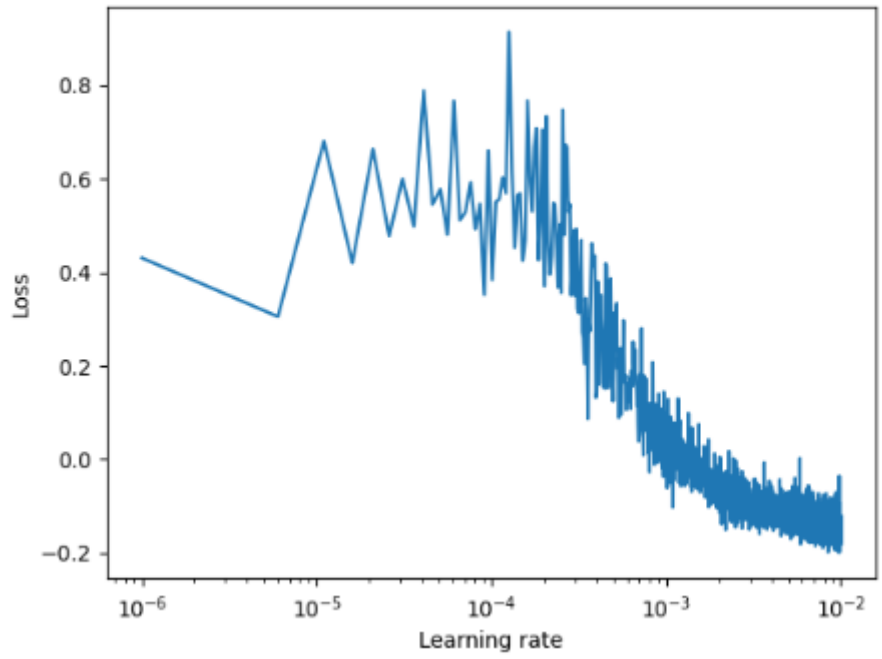
实验结果

中文标注和英文标注

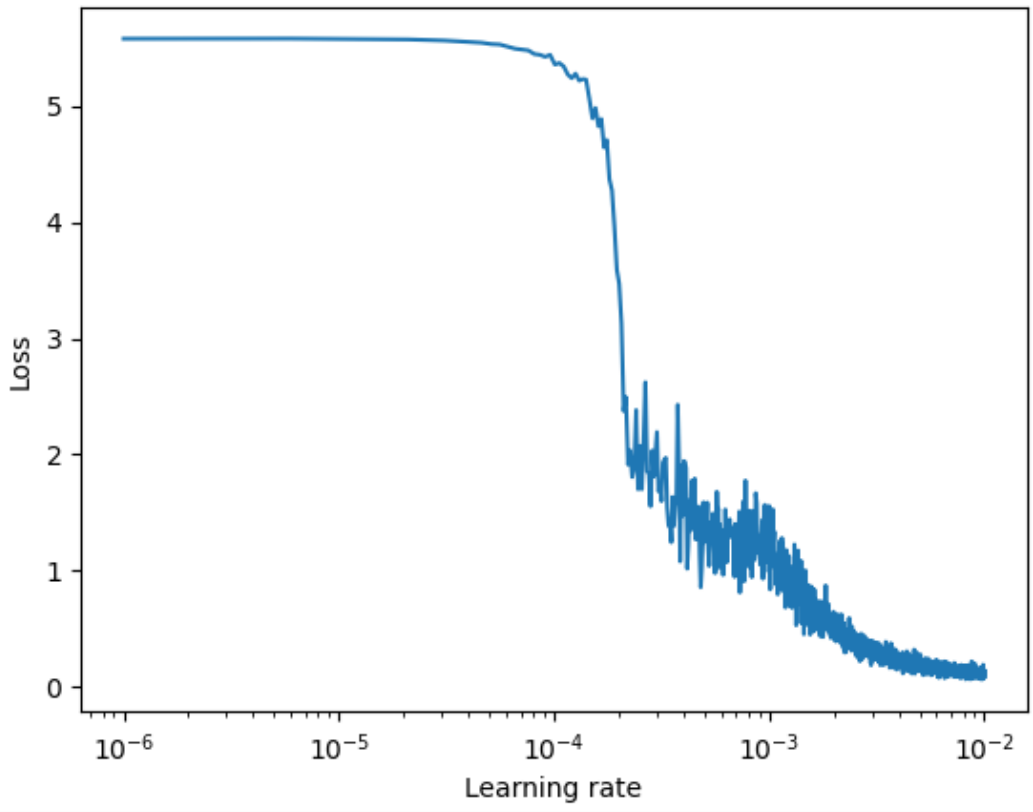
实验结果在实验步骤中已经详细列出，在此不再赘述。

CRF

- 读取预训练模型开始训练，损失函数变化如下



- 新建模型开始训练，损失函数变化如下



- loss=0.18参数结果

```
('华为', 'a') ('是', 'vshi') ('全', 'a') ('球', 'q') ('领先', 'vi')
('的', 'ude1') ('I', 'ng') ('C', 'x') ('T', 'w') ('（', 'w')
('信'息', 'c') ('与', 'qv') ('通信', 'vn') ('）', 'w') ('基础', 'n')
('设施', 'n') ('和', 'cc') ('智', 'n') ('能', 'v') ('终', 'd')
('端'v') ('供', 'x') ('商', 'w') ('，', 'w') ('致力于', 'v') ('把',
'pba') ('数字', 'n') ('世界', 'vi') ('带入', 'v') ('每', 'n') ('个
人') ('每', 'n') ('个', 'q') ('家庭', 'nz') ('、', 'w') ('每', 'n')
('个', 'q') ('组织', 'n') ('，', 'w') ('构', 'd') ('建', 'v') ('万,
'nr') ('联', 'ng') ('的', 'ude1') ('智', 'n') ('能', 'v') ('世界',
'vi') ('。', 'w') ('我们', 'rr') ('在', 'p') ('通信', 'vn') ('网',
'w') ('I', 'ng') ('T', 'w') ('、', 'w') ('智', 'n') ('能', 'v')
('终', 'd') ('端', 'v') ('和', 'cc') ('云', 'b') ('服务', 'vn')
(ng') ('领域', 'vi') ('为', 'p') ('客户', 'n') ('提', 'v') ('供',
'x') ('有', 'vyou') ('竞', 'bl') ('争', 'v') ('力', 'n') ('、',
'w'') ('可信赖', 'nz') ('的', 'ude1') ('产品', 'n') ('、', 'w') ('解
决方案与', 'gi') ('服务', 'vn') ('，', 'w') ('与', 'qv') ('生态
伙',z') ('伴', 'c') ('开放', 'v') ('合作', 'vn') ('，持', 'ntu')
('续', 'vd') ('为', 'p') ('客户', 'n') ('创造价值', 'nz') ('，',
'w') ( 'v') ('个人潜', 'n') ('能', 'v') ('，', 'w') ('丰富', 'ns')
('家庭生活', 'nz') ('，', 'w') ('激发', 'v') ('组织', 'n') ('创新',
'vi坚', 'a') ('持围', 'v') ('绕客户', 'n') ('需求', 'n') ('持续创新',
'nz') ('，', 'w') ('加大', 'v') ('基', 'ng') ('础', 'ag') ('研究'
('，', 'w') ('厚积薄发', 'vl') ('，', 'w') ('推动', 'v') ('世界进步',
'nz') ('。', 'w') ('华为', 'a') ('成立', 'vi') ('于', 'p') (' 'w')
('是', 'vshi') ('一家', 'mq') ('由', 'p') ('员工', 'n') ('持有',
'v') ('全部', 'm') ('股', 'c') ('份', 'q') ('的', 'ude1') ('民,
'qt') ('企', 'ng') ('业', 'm') ('，', 'w') ('目前', 't') ('有',
'vyou') ('1', 'm') ('8', 'p') ('万', 'w') ('员工', 'n') ('，',
'w'mq') ('遍及', 'v') ('170多个', 'mq') ('国', 'n') ('家', 'p')
('和', 'cc') ('地区', 'n') ('。', 'w')
```

- loss=0.03参数结果

```
('华为', 'ns') ('是', 'vshi') ('全球', 'n') ('领先', 'vi') ('的',
'ude1') ('ICT', 'x') ('（', 'w') ('信息', 'n') ('与', 'cc') ('通信
'w') ('基础设施', 'gi') ('和', 'cc') ('智能', 'n') ('终端', 'n')
('提', 'v') ('供商', 'vn') ('，', 'w') ('致力于', 'v') ('把', 'pba世
界', 'n') ('带入', 'v') ('每个人', 'mq') ('、', 'w') ('每个', 'r')
('家庭', 'n') ('、', 'w') ('每个', 'r') ('组织', 'n') ('，', 'w物互
联', 'nz') ('的', 'ude1') ('智能世界', 'nz') ('。', 'w') ('我们',
'rr') ('在', 'p') ('通信', 'vn') ('网络', 'n') ('、', 'w') ('I')
('智能终端', 'nz') ('和', 'cc') ('云', 'vg') ('服务', 'vn') ('等',
'udeng') ('领域', 'n') ('为', 'p') ('客户', 'n') ('提供', 'v')'竞争
力', 'n') ('、', 'w') ('安全', 'an') ('可', 'v') ('信赖', 'vn')
('的', 'ude1') ('产品', 'n') ('、', 'w') ('解决方案', 'gi') ('与vn')
('，', 'w') ('与', 'cc') ('生态', 'n') ('伙伴', 'n') ('开放', 'v')
('合作', 'vn') ('，', 'w') ('持续', 'vd') ('为', 'p') ('客户v') ('价
值', 'n') ('，', 'w') ('释放', 'v') ('个人', 'n') ('潜能', 'n')
('，', 'w') ('丰富', 'a') ('家庭生活', 'nz') ('，', 'w') ('激', 'v')
('组织', 'n') ('创新', 'vi') ('。', 'w') ('华为', 'ns') ('坚持',
'v') ('围绕', 'v') ('客户', 'n') ('需求', 'n') ('持续', 'vd', ',
'w') ('加大', 'v') ('基础', 'n') ('研究', 'vn') ('投入', 'v') ('，',
'w') ('厚积薄发', 'nz') ('，', 'w') ('推动', 'v') ('世界' ('。',
'w') ('华为', 'ns') ('成立', 'vi') ('于', 'p') ('1987年', 't')
('，', 'w') ('是', 'vshi') ('一家', 'n') ('由', 'p') ('员工',v')
('全部', 'm') ('股份', 'n') ('的', 'ude1') ('民营企业', 'nz') ('，',
'w') ('目前', 't') ('有', 'vyou') ('18万', 'm') ('员工', 'n业务',
'n') ('遍及', 'v') ('170多个', 'mq') ('国家', 'n') ('和', 'cc') ('地
区', 'n') ('。', 'w')
```

实验总结

通过观察，我认为和上次实验类似，中文的词性标注是难于英文的。

英文词性标注

以上三种实施的英文分词方法都有一个共同的缺点：没有将名词进一步划分。他们大多将名词分成（普通）名词和专有名词，而没有进一步考虑，该名词是否是人名，地名等。

- `Spacy` 相较于其他两种方法，最突出的特点是能够对标点进行标注。整体上，词性标注效果十分不错。其标注词性的分类与另外两种不太相同
- `StanfordCoreNLP` 与 `NLTK` 的标注方式是一致的。 `NLTK` 中一些明显的错误，在 `StanfordCoreNLP` 中没有出现， `StanfordCoreNLP` 的词性标注效果明显优于 `NLTK` ，尽管 `NLTK` 可以完成分词的任务。

中文词性标注

- `jieba` 中的 `paddle` 模式，对中文词性标注的效果很好，可以区分不同专有名词（分为地点，时间，人名等），且其标注错误比较少，在以上中文词性标注方法中效果是最好的。但是对标点的分词比较混乱。 `jieba` 的默认模式标注效果明显不如 `paddle` 模式，标注比较混乱。
- `StanfordCoreNLP` 可以对标点进行标注，但是对名词的划分不够细致，且词性标注错误比较多。
- `SnowNLP` 可以基本完成标注任务，但是相较于其他方法没有明显的优点。其分词效果不好严重影响了标注效果。
- `THULAC` 标注效果不错，可以区分人名，机构名，量词等，总体效果十分不错。但是其分词时太细致，一定程度上影响标注效果。
- `NLPIR` 的词性标注方式为正常的英文，而不是自定义的词性表，看起来非常易懂。但是没有对名词进行区分，和其他方法相比，甚至没有将名词和专有名词区分开。词性标注比较乱。

CRF

能够看出一定的分词和词性标注能力。但是，由于训练不完全，分词和词性标注都奇奇怪怪的😃。

关于环境配置问题，详见博客[山东大学nlp实验--CRF环境配置_长命百岁的博客-CSDN博客](#)（本人）