中文处理FoolNLTK：

1. 中文处理工具，
4. 可能不是最快的开源中文分词，但很可能是最准的开源中文分词
2. 提供BiLSTM模型来分词。

双向LSTM相当于两个LSTM，一个正向输入序列，一个反向输入序列，再将两者的输出结合起来作为最终的结果。

解决的问题：可以做到双向分析分本进行分词。

如： 研究生命的起源

正向：研究生/命/的/起源

逆向：研究/生命/的/起源

4. 特点：基于BiLSTM模型来训练、准确度高
5. 可用户自定义字典、支持自训练、允许batch处理

代码分析

```
import fool:
fool里有初始定义文件如init.py
[load_model: 加载模型
 cut: 分词
 pos_cut: 词性分析
 ner: 实体识别
 analysis: 分析
 load_userdict: 加载自定义辞典
 delete_userdict: 删除辞典
```

1. fool.cut( ) : 分词

```
text = "一个傻子在北京"
print(fool.cut(text))
```

输出：
[['一个', '傻子', '在', '北京']]

2. load_userdict('path') 加载自定义辞典

```
#用户可自定义辞典，词的权重越高，词的长度越长就越越可能出现，  权重值请大于1
fool.load_userdict('/Users/jiweilu/Desktop/1.txt')
text = ["我在北京天安门看你难受香菇", "我在北京晒太阳你在非洲看雪"]

print(fool.cut(text))
```

辞典一

```
难受香菇 10
难受 5
香菇 5
什么鬼 10
分词工具 10
北京 10
北京天安门 10
```

词典二

```
难受香菇 10
难受 20
香菇 15
什么鬼 10
分词工具 10
北京 10
北京天安门 10
```

输出1：
[['我', '在', '北京天安门', '看', '你', '难受香菇'], ['我', '在', '北京', '晒太阳', '你', '在', '非洲', '看', '雪']]

输出2:
[['我', '在', '北京天安门', '看', '你', '难受', '香菇'], ['我', '在', '北京', '晒太阳', '你', '在', '非洲', '看', '雪']]

3. pos_cut( )定义词性

n/名词 np/人名 ns/地名 ni/机构名 nz/其它专名
m/数词 q/量词 mq/数量词 t/时间词 f/方位词 s/处所词
v/动词 a/形容词 d/副词 h/前接成分 k/后接成分
i/习语 j/简称 r/代词 c/连词 p/介词 u/助词 y/语气助词
e/叹词 o/拟声词 g/语素 w/标点 x/其它

```
text = ["一个傻子在北京"]
print(fool.pos_cut(text))
```

输出：[[('一个', 'm'), ('傻子', 'n'), ('在', 'p'), ('北京', 'ns')]]

4. analysis(text)( ) 实体识别

```
text = ["一个傻子在北京","你好啊"]
words,ners = fool.analysis(text)
print(ners)
```

```
def analysis(self, text_list):
    words = self.cut(text_list)
    pos_labels = self.pos(words)
    ners = self.ner(text_list)
    word_inf = [list(zip(ws, ps)) for ws, ps in zip(words, pos_labels)]
    return word_inf, ners
```

输出：

#[[(5, 8, 'location', '北京')]]

5.fool.delete_userdict();

更多应用：https://www.kesci.com/home/project/5b863f1131902f000f64adce