# jieba

“结巴”中文分词：做最好的 Python 中文分词组件

"Jieba" (Chinese for "to stutter") Chinese text segmentation: built to be the best Python Chinese word segmentation module.

* *Scroll down for English documentation.*

# 特点

* 支持四种分词模式：
  + 精确模式，试图将句子最精确地切开，适合文本分析；
  + 全模式，把句子中所有的可以成词的词语都扫描出来, 速度非常快，但是不能解决歧义；
  + 搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。
  + paddle模式，利用PaddlePaddle深度学习框架，训练序列标注（双向GRU）网络模型实现分词。同时支持词性标注。paddle模式使用需安装paddlepaddle-tiny，pip install paddlepaddle-tiny==1.6.1。目前paddle模式支持jieba v0.40及以上版本。jieba v0.40以下版本，请升级jieba，pip install jieba --upgrade 。[PaddlePaddle官网](https://www.paddlepaddle.org.cn/)
* 支持繁体分词
* 支持自定义词典
* MIT 授权协议

# 安装说明

代码对 Python 2/3 均兼容

* 全自动安装：easy\_install jieba 或者 pip install jieba / pip3 install jieba
* 半自动安装：先下载 http://pypi.python.org/pypi/jieba/ ，解压后运行 python setup.py install
* 手动安装：将 jieba 目录放置于当前目录或者 site-packages 目录
* 通过 import jieba 来引用
* 如果需要使用paddle模式下的分词和词性标注功能，请先安装paddlepaddle-tiny，pip install paddlepaddle-tiny==1.6.1。

# 算法

* 基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)
* 采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合
* 对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法

# 主要功能

1. 分词

* jieba.cut 方法接受四个输入参数: 需要分词的字符串；cut*all 参数用来控制是否采用全模式；HMM 参数用来控制是否使用 HMM 模型；use*paddle 参数用来控制是否使用paddle模式下的分词模式，paddle模式采用延迟加载方式，通过enable\_paddle接口安装paddlepaddle-tiny，并且import相关代码；
* jieba.cut\_for\_search 方法接受两个参数：需要分词的字符串；是否使用 HMM 模型。该方法适合用于搜索引擎构建倒排索引的分词，粒度比较细
* 待分词的字符串可以是 unicode 或 UTF-8 字符串、GBK 字符串。注意：不建议直接输入 GBK 字符串，可能无法预料地错误解码成 UTF-8
* jieba.cut 以及 jieba.cut\_for\_search 返回的结构都是一个可迭代的 generator，可以使用 for 循环来获得分词后得到的每一个词语(unicode)，或者用
* jieba.lcut 以及 jieba.lcut\_for\_search 直接返回 list
* jieba.Tokenizer(dictionary=DEFAULT\_DICT) 新建自定义分词器，可用于同时使用不同词典。jieba.dt 为默认分词器，所有全局分词相关函数都是该分词器的映射。

代码示例

# encoding=utf-8  
import jieba  
  
jieba.enable\_paddle()# 启动paddle模式。 0.40版之后开始支持，早期版本不支持  
strs=["我来到北京清华大学","乒乓球拍卖完了","中国科学技术大学"]  
for str in strs:  
 seg\_list = jieba.cut(str,use\_paddle=True) # 使用paddle模式  
 print("Paddle Mode: " + '/'.join(list(seg\_list)))  
  
seg\_list = jieba.cut("我来到北京清华大学", cut\_all=True)  
print("Full Mode: " + "/ ".join(seg\_list)) # 全模式  
  
seg\_list = jieba.cut("我来到北京清华大学", cut\_all=False)  
print("Default Mode: " + "/ ".join(seg\_list)) # 精确模式  
  
seg\_list = jieba.cut("他来到了网易杭研大厦") # 默认是精确模式  
print(", ".join(seg\_list))  
  
seg\_list = jieba.cut\_for\_search("小明硕士毕业于中国科学院计算所，后在日本京都大学深造") # 搜索引擎模式  
print(", ".join(seg\_list))

输出:

【全模式】: 我/ 来到/ 北京/ 清华/ 清华大学/ 华大/ 大学  
  
【精确模式】: 我/ 来到/ 北京/ 清华大学  
  
【新词识别】：他, 来到, 了, 网易, 杭研, 大厦 (此处，“杭研”并没有在词典中，但是也被Viterbi算法识别出来了)  
  
【搜索引擎模式】： 小明, 硕士, 毕业, 于, 中国, 科学, 学院, 科学院, 中国科学院, 计算, 计算所, 后, 在, 日本, 京都, 大学, 日本京都大学, 深造

1. 添加自定义词典

### 载入词典

* 开发者可以指定自己自定义的词典，以便包含 jieba 词库里没有的词。虽然 jieba 有新词识别能力，但是自行添加新词可以保证更高的正确率
* 用法： jieba.load*userdict(file*name) # file\_name 为文件类对象或自定义词典的路径
* 词典格式和 dict.txt 一样，一个词占一行；每一行分三部分：词语、词频（可省略）、词性（可省略），用空格隔开，顺序不可颠倒。file\_name 若为路径或二进制方式打开的文件，则文件必须为 UTF-8 编码。
* 词频省略时使用自动计算的能保证分出该词的词频。

**例如：**

创新办 3 i  
云计算 5  
凱特琳 nz  
台中

* 更改分词器（默认为 jieba.dt）的 tmp\_dir 和 cache\_file 属性，可分别指定缓存文件所在的文件夹及其文件名，用于受限的文件系统。
* 范例：
  + 自定义词典：https://github.com/fxsjy/jieba/blob/master/test/userdict.txt
  + 用法示例：https://github.com/fxsjy/jieba/blob/master/test/test\_userdict.py
    - 之前： 李小福 / 是 / 创新 / 办 / 主任 / 也 / 是 / 云 / 计算 / 方面 / 的 / 专家 /
    - 加载自定义词库后： 李小福 / 是 / 创新办 / 主任 / 也 / 是 / 云计算 / 方面 / 的 / 专家 /

### 调整词典

* 使用 add\_word(word, freq=None, tag=None) 和 del\_word(word) 可在程序中动态修改词典。
* 使用 suggest\_freq(segment, tune=True) 可调节单个词语的词频，使其能（或不能）被分出来。
* 注意：自动计算的词频在使用 HMM 新词发现功能时可能无效。

代码示例：

>>> print('/'.join(jieba.cut('如果放到post中将出错。', HMM=False)))  
如果/放到/post/中将/出错/。  
>>> jieba.suggest\_freq(('中', '将'), True)  
494  
>>> print('/'.join(jieba.cut('如果放到post中将出错。', HMM=False)))  
如果/放到/post/中/将/出错/。  
>>> print('/'.join(jieba.cut('「台中」正确应该不会被切开', HMM=False)))  
「/台/中/」/正确/应该/不会/被/切开  
>>> jieba.suggest\_freq('台中', True)  
69  
>>> print('/'.join(jieba.cut('「台中」正确应该不会被切开', HMM=False)))  
「/台中/」/正确/应该/不会/被/切开

* "通过用户自定义词典来增强歧义纠错能力" --- https://github.com/fxsjy/jieba/issues/14

1. 关键词提取

### 基于 TF-IDF 算法的关键词抽取

import jieba.analyse

* jieba.analyse.extract\_tags(sentence, topK=20, withWeight=False, allowPOS=())
  + sentence 为待提取的文本
  + topK 为返回几个 TF/IDF 权重最大的关键词，默认值为 20
  + withWeight 为是否一并返回关键词权重值，默认值为 False
  + allowPOS 仅包括指定词性的词，默认值为空，即不筛选
* jieba.analyse.TFIDF(idf*path=None) 新建 TFIDF 实例，idf*path 为 IDF 频率文件

代码示例 （关键词提取）

https://github.com/fxsjy/jieba/blob/master/test/extract\_tags.py

关键词提取所使用逆向文件频率（IDF）文本语料库可以切换成自定义语料库的路径

* 用法： jieba.analyse.set*idf*path(file*name) # file*name为自定义语料库的路径
* 自定义语料库示例：https://github.com/fxsjy/jieba/blob/master/extra\_dict/idf.txt.big
* 用法示例：https://github.com/fxsjy/jieba/blob/master/test/extract*tags*idfpath.py

关键词提取所使用停止词（Stop Words）文本语料库可以切换成自定义语料库的路径

* 用法： jieba.analyse.set*stop*words(file*name) # file*name为自定义语料库的路径
* 自定义语料库示例：https://github.com/fxsjy/jieba/blob/master/extra*dict/stop*words.txt
* 用法示例：https://github.com/fxsjy/jieba/blob/master/test/extract*tags*stop\_words.py

关键词一并返回关键词权重值示例

* 用法示例：https://github.com/fxsjy/jieba/blob/master/test/extract*tags*with\_weight.py

### 基于 TextRank 算法的关键词抽取

* jieba.analyse.textrank(sentence, topK=20, withWeight=False, allowPOS=('ns', 'n', 'vn', 'v')) 直接使用，接口相同，注意默认过滤词性。
* jieba.analyse.TextRank() 新建自定义 TextRank 实例

算法论文： [TextRank: Bringing Order into Texts](http://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf)

#### 基本思想:

1. 将待抽取关键词的文本进行分词
2. 以固定窗口大小(默认为5，通过span属性调整)，词之间的共现关系，构建图
3. 计算图中节点的PageRank，注意是无向带权图

#### 使用示例:

见 [test/demo.py](https://github.com/fxsjy/jieba/blob/master/test/demo.py)

1. 词性标注

* jieba.posseg.POSTokenizer(tokenizer=None) 新建自定义分词器，tokenizer 参数可指定内部使用的 jieba.Tokenizer 分词器。jieba.posseg.dt 为默认词性标注分词器。
* 标注句子分词后每个词的词性，采用和 ictclas 兼容的标记法。
* 除了jieba默认分词模式，提供paddle模式下的词性标注功能。paddle模式采用延迟加载方式，通过enable\_paddle()安装paddlepaddle-tiny，并且import相关代码；
* 用法示例

>>> import jieba  
>>> import jieba.posseg as pseg  
>>> words = pseg.cut("我爱北京天安门") #jieba默认模式  
>>> jieba.enable\_paddle() #启动paddle模式。 0.40版之后开始支持，早期版本不支持  
>>> words = pseg.cut("我爱北京天安门",use\_paddle=True) #paddle模式  
>>> for word, flag in words:  
... print('%s %s' % (word, flag))  
...  
我 r  
爱 v  
北京 ns  
天安门 ns

paddle模式词性标注对应表如下：

paddle模式词性和专名类别标签集合如下表，其中词性标签 24 个（小写字母），专名类别标签 4 个（大写字母）。

|  |
| --- |
| 标签 |
| n |
| nr |
| nz |
| a |
| m |
| c |
| PER |

1. 并行分词

* 原理：将目标文本按行分隔后，把各行文本分配到多个 Python 进程并行分词，然后归并结果，从而获得分词速度的可观提升
* 基于 python 自带的 multiprocessing 模块，目前暂不支持 Windows
* 用法：
  + jieba.enable\_parallel(4) # 开启并行分词模式，参数为并行进程数
  + jieba.disable\_parallel() # 关闭并行分词模式
* 例子：https://github.com/fxsjy/jieba/blob/master/test/parallel/test\_file.py
* 实验结果：在 4 核 3.4GHz Linux 机器上，对金庸全集进行精确分词，获得了 1MB/s 的速度，是单进程版的 3.3 倍。
* **注意**：并行分词仅支持默认分词器 jieba.dt 和 jieba.posseg.dt。

1. Tokenize：返回词语在原文的起止位置

* 注意，输入参数只接受 unicode
* 默认模式

result = jieba.tokenize(u'永和服装饰品有限公司')  
for tk in result:  
 print("word %s\t\t start: %d \t\t end:%d" % (tk[0],tk[1],tk[2]))  
word 永和 start: 0 end:2  
word 服装 start: 2 end:4  
word 饰品 start: 4 end:6  
word 有限公司 start: 6 end:10

* 搜索模式

result = jieba.tokenize(u'永和服装饰品有限公司', mode='search')  
for tk in result:  
 print("word %s\t\t start: %d \t\t end:%d" % (tk[0],tk[1],tk[2]))  
word 永和 start: 0 end:2  
word 服装 start: 2 end:4  
word 饰品 start: 4 end:6  
word 有限 start: 6 end:8  
word 公司 start: 8 end:10  
word 有限公司 start: 6 end:10

1. ChineseAnalyzer for Whoosh 搜索引擎

* 引用： from jieba.analyse import ChineseAnalyzer
* 用法示例：https://github.com/fxsjy/jieba/blob/master/test/test\_whoosh.py

1. 命令行分词

使用示例：python -m jieba news.txt > cut\_result.txt

命令行选项（翻译）：

使用: python -m jieba [options] filename  
  
结巴命令行界面。  
  
固定参数:  
 filename 输入文件  
  
可选参数:  
 -h, --help 显示此帮助信息并退出  
 -d [DELIM], --delimiter [DELIM]  
 使用 DELIM 分隔词语，而不是用默认的' / '。  
 若不指定 DELIM，则使用一个空格分隔。  
 -p [DELIM], --pos [DELIM]  
 启用词性标注；如果指定 DELIM，词语和词性之间  
 用它分隔，否则用 \_ 分隔  
 -D DICT, --dict DICT 使用 DICT 代替默认词典  
 -u USER\_DICT, --user-dict USER\_DICT  
 使用 USER\_DICT 作为附加词典，与默认词典或自定义词典配合使用  
 -a, --cut-all 全模式分词（不支持词性标注）  
 -n, --no-hmm 不使用隐含马尔可夫模型  
 -q, --quiet 不输出载入信息到 STDERR  
 -V, --version 显示版本信息并退出  
  
如果没有指定文件名，则使用标准输入。

--help 选项输出：

$> python -m jieba --help  
Jieba command line interface.  
  
positional arguments:  
 filename input file  
  
optional arguments:  
 -h, --help show this help message and exit  
 -d [DELIM], --delimiter [DELIM]  
 use DELIM instead of ' / ' for word delimiter; or a  
 space if it is used without DELIM  
 -p [DELIM], --pos [DELIM]  
 enable POS tagging; if DELIM is specified, use DELIM  
 instead of '\_' for POS delimiter  
 -D DICT, --dict DICT use DICT as dictionary  
 -u USER\_DICT, --user-dict USER\_DICT  
 use USER\_DICT together with the default dictionary or  
 DICT (if specified)  
 -a, --cut-all full pattern cutting (ignored with POS tagging)  
 -n, --no-hmm don't use the Hidden Markov Model  
 -q, --quiet don't print loading messages to stderr  
 -V, --version show program's version number and exit  
  
If no filename specified, use STDIN instead.

## 延迟加载机制

jieba 采用延迟加载，import jieba 和 jieba.Tokenizer() 不会立即触发词典的加载，一旦有必要才开始加载词典构建前缀字典。如果你想手工初始 jieba，也可以手动初始化。

import jieba  
jieba.initialize() # 手动初始化（可选）

在 0.28 之前的版本是不能指定主词典的路径的，有了延迟加载机制后，你可以改变主词典的路径:

jieba.set\_dictionary('data/dict.txt.big')

例子： https://github.com/fxsjy/jieba/blob/master/test/test*change*dictpath.py

# 其他词典

1. 占用内存较小的词典文件 https://github.com/fxsjy/jieba/raw/master/extra\_dict/dict.txt.small
2. 支持繁体分词更好的词典文件 https://github.com/fxsjy/jieba/raw/master/extra\_dict/dict.txt.big

下载你所需要的词典，然后覆盖 jieba/dict.txt 即可；或者用 jieba.set\_dictionary('data/dict.txt.big')

# 其他语言实现

## 结巴分词 Java 版本

作者：piaolingxue 地址：https://github.com/huaban/jieba-analysis

## 结巴分词 C++ 版本

作者：yanyiwu 地址：https://github.com/yanyiwu/cppjieba

## 结巴分词 Rust 版本

作者：messense, MnO2 地址：https://github.com/messense/jieba-rs

## 结巴分词 Node.js 版本

作者：yanyiwu 地址：https://github.com/yanyiwu/nodejieba

## 结巴分词 Erlang 版本

作者：falood 地址：https://github.com/falood/exjieba

## 结巴分词 R 版本

作者：qinwf 地址：https://github.com/qinwf/jiebaR

## 结巴分词 iOS 版本

作者：yanyiwu 地址：https://github.com/yanyiwu/iosjieba

## 结巴分词 PHP 版本

作者：fukuball 地址：https://github.com/fukuball/jieba-php

## 结巴分词 .NET(C#) 版本

作者：anderscui 地址：https://github.com/anderscui/jieba.NET/

## 结巴分词 Go 版本

* 作者: wangbin 地址: https://github.com/wangbin/jiebago
* 作者: yanyiwu 地址: https://github.com/yanyiwu/gojieba

## 结巴分词Android版本

* 作者 Dongliang.W 地址：https://github.com/452896915/jieba-android

# 友情链接

* https://github.com/baidu/lac 百度中文词法分析（分词+词性+专名）系统
* https://github.com/baidu/AnyQ 百度FAQ自动问答系统
* https://github.com/baidu/Senta 百度情感识别系统

# 系统集成

1. Solr: https://github.com/sing1ee/jieba-solr

# 分词速度

* 1.5 MB / Second in Full Mode
* 400 KB / Second in Default Mode
* 测试环境: Intel(R) Core(TM) i7-2600 CPU @ 3.4GHz；《围城》.txt

# 常见问题

## 1. 模型的数据是如何生成的？

详见： https://github.com/fxsjy/jieba/issues/7

## 2. “台中”总是被切成“台 中”？（以及类似情况）

P(台中) ＜ P(台)×P(中)，“台中”词频不够导致其成词概率较低

解决方法：强制调高词频

jieba.add\_word('台中')` 或者 `jieba.suggest\_freq('台中', True)

## 3. “今天天气 不错”应该被切成“今天 天气 不错”？（以及类似情况）

解决方法：强制调低词频

jieba.suggest\_freq(('今天', '天气'), True)

或者直接删除该词 jieba.del\_word('今天天气')

## 4. 切出了词典中没有的词语，效果不理想？

解决方法：关闭新词发现

jieba.cut('丰田太省了', HMM=False)` `jieba.cut('我们中出了一个叛徒', HMM=False)

**更多问题请点击**：https://github.com/fxsjy/jieba/issues?sort=updated&state=closed

# 修订历史

https://github.com/fxsjy/jieba/blob/master/Changelog