



NLP项目

Jieba&HanLP&jiagu工具

NLP常用工具

⌚ jieba:

⌚ <https://github.com/fxsjy/jieba>

⌚ HanLP:

⌚ <http://hanlp.com/>

⌚ <https://github.com/hankcs/pyhanlp>

⌚ jiagu

⌚ <https://github.com/ownthink/Jiagu>

⌚ gensim

⌚ <https://radimrehurek.com/gensim/>

⌚ ltp:

⌚ <http://ltp.ai/demo.html>

⌚ <https://github.com/HIT-SCIR/ltp>

NLP基础 分词

⌚ 分词是指将文本数据转换为一个一个的单词，是中文的NLP自然语言处理过程中的基础；因为对于文本信息来讲，我们可以认为文本中的单词可以体现文本的特征信息，所以在进行自然语言相关任务的时候，第一步操作就是需要将文本信息转换为单词序列，使用单词序列来表达文本的特征信息。

⌚ 分词：**通过某种技术将连续的文本分隔成更具有语言语义学上意义的“词”**。这个过程就叫做分词。

⌚ Python中汉字分词包：jieba

⌚ 安装方式： `pip install jieba`

⌚ Github: <https://github.com/fxsjy/jieba>

```
(base) C:\Users\gerry_17578261252713>pip install jieba
Looking in indexes: https://mirrors.aliyun.com/pypi/simple
Collecting jieba
Installing collected packages: jieba
Successfully installed jieba-0.42.1

(base) C:\Users\gerry_17578261252713>python
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64
Type "help", "copyright", "credits" or "license" for more in
>>> import jieba
>>> jieba.__version__
'0.42.1'
```

NLP基础_分词_jieba

- ⌚ **"Jieba"** Chinese text segmentation: built to be the best Python Chinese word segmentation module.
- ⌚ Jieba常用的一种Python语言的**中文分词**和**词性标注**工具；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，然后采用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/新词(在词典中未找到)的中文分词和词性标注问题使用HMM的Viterbi算法来进行构造。

NLP基础_分词_jieba

⌚ 分词模式:

⌚ 精确模式

⌚ `jieba.cut(str)`

⌚ 试图将句子最精确地切开, 适合文本分析;

⌚ 全模式

⌚ `jieba.cut(str, cut_all=True)`

⌚ 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

⌚ `jieba.cut_for_search(str)`

⌚ 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。

NLP基础_分词_jieba

⌚ 基础功能:

⌚ 1. 分词

⌚ **前缀字典匹配**、HMM模型Viterbi算法

⌚ 2. 自定义词典添加

⌚ 3. 关键词抽取

⌚ TF-IDF、TextRank

⌚ 4. 词性标注

⌚ HMM模型Viterbi算法

⌚ 5. 并行分词

⌚ 当前版本不支持windows

NLP基础_分词

⌚ 常规的分词技术:

⌚ 规则分词

⌚ 通过维护词典，通过切分语句的时候，将语句的每个字符串与词表中的词进行逐一匹配，找到则切分，否则不予切分。主要包括：

⌚ 正向最大匹配法

⌚ 逆向最大匹配法

⌚ 双向最大匹配法

⌚ 统计分词

⌚ 通过统计各个词在训练文本中出现的次数得到词的可信度，当连续的各个字出现的频度超过某个值的时候，就可以认为这个连续的各个字属于一个词。主要包括：

⌚ n-gram模型

⌚ HMM、CRF、LSTM、BERT

⌚ 混合分词

HanLP

⌚ 自然语言处理， 功能如下：

⌚ **中文分词**

⌚ **词性标注**

⌚ **命名实体识别**

⌚ 依存句法分析

⌚ 新词发现

⌚ 关键词短语提取

⌚ 自动摘要

⌚ 文本分类聚类

⌚ 拼音简繁

- <http://hanlp.com/>
- <https://github.com/hankcs/HanLP>
- <https://github.com/hankcs/pyhanlp>
- <http://www.hankcs.com/nlp/part-of-speech-tagging.html>
- <https://hanlp.hankcs.com/demos/pos.html>

HanLP

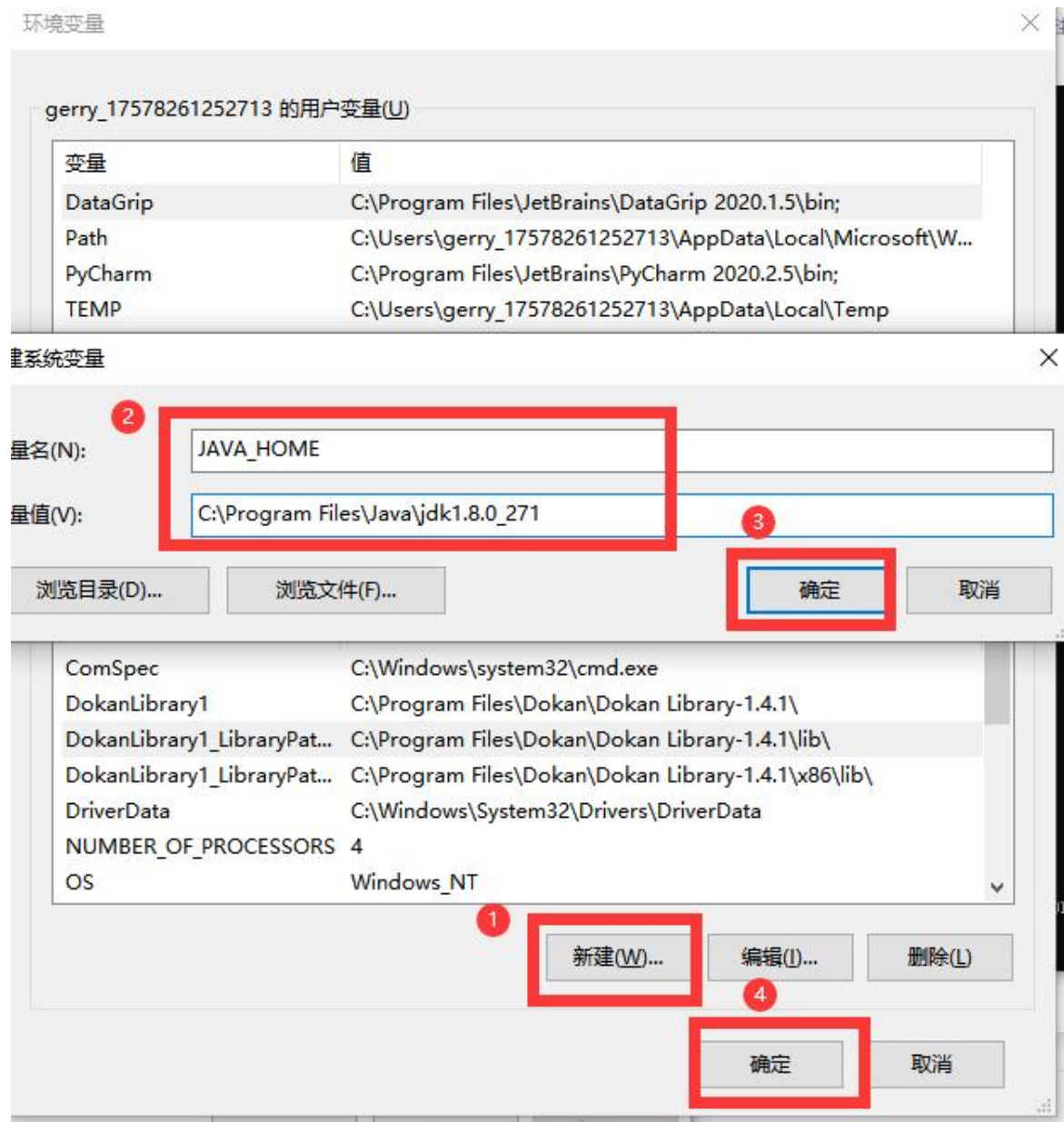
⌚ 安装过程(版本是1.x):

- ⌚ 1. 安装JDK1.8以上的版本, 配置JAVA_HOME和PATH环境变量;
- ⌚ 2. `pip install pyhanlp`安装PyHanLP;
- ⌚ 3. 下载data-for-1.7.5.zip, 并将其解压放置在pyhanlp模块下的static文件夹下的data子文件夹中。
- ⌚ 4. 进入Python命令行, 执行`import pyhanlp`(自动下载配置相关资源); 如果是网络原因, 可以直接下载hanlp-1.7.5-release.zip解压后将文件放置在static文件夹下。
- ⌚ 5. 配置pyhanlp(配置hanlp.properties)。
- ⌚ NOTE: 下载路径: <https://github.com/hankcs/HanLP/releases>
- ⌚ NOTE: python版本仅支持3.7或者3.8.

HanLP安装 1.x



HanLP安装 1.x



HanLP安装 1.x

```
(base) C:\Users\gerry_17578261252713>pip install pyhanlp
Looking in indexes: https://mirrors.aliyun.com/pypi/simple
Collecting pyhanlp
Collecting jpype1==0.7.0 (from pyhanlp)
  Using cached https://mirrors.aliyun.com/pypi/packages/d3/08/f4bb58c1c0dff93e96
d1/JPyPe1-0.7.0-cp37-cp37m-win_amd64.whl
Collecting hanlp-downloader (from pyhanlp)
Requirement already satisfied: requests in d:\anaconda3\lib\site-packages (from
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in d:\anaconda3\lib\site-pa
>pyhanlp) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in d:\anaconda3\lib\site-packa
hanlp) (2019.9.11)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in d:\ana
>hanlp-downloader->pyhanlp) (1.24.2)
Requirement already satisfied: idna<2.9,>=2.5 in d:\anaconda3\lib\site-packages
p) (2.8)
Installing collected packages: jpype1, hanlp-downloader, pyhanlp
Successfully installed hanlp-downloader-0.0.25 jpype1-0.7.0 pyhanlp-0.1.84
```

安装jpype如果失败，需要单独安装

HanLP安装 1.x

```
(base) C:\Users\germy_17578261252713>python
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyhanlp
下载 https://file.hankcs.com/hanlp/hanlp-1.8.3-release.zip 到 D:\anaconda3\lib\site-packages\pyhanlp\static\hanlp-1.8.3-release.zip
100% 1.8 MiB 1.8 MiB/s ETA: 0 [=====]
下载 https://file.hankcs.com/hanlp/data-for-1.7.5.zip 到 D:\anaconda3\lib\site-packages\pyhanlp\static\data-for-1.8.3.zip
下载失败 https://file.hankcs.com/hanlp/data-for-1.7.5.zip 由于 ConnectionError(MaxRetryError("HTTPConnectionPool(host='file.hankcs.workers.dev', port=443): Max retries exceeded with url: /hanlp/data-for-1.7.5.zip (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x000001856FF05DC8>: Failed to establish a new connection: [WinError 10060] 由于连接方在一段时间后没有正确答复或连接的主机没有反应, 连接尝试失败。'))"))
请参考 https://od.hankcs.com/book/intro_nlp/ 执行手动安装.
或手动下载 https://file.hankcs.com/hanlp/data-for-1.7.5.zip 到 D:\anaconda3\lib\site-packages\pyhanlp\static\data-for-1.8.3.zip
是否前往 https://od.hankcs.com/book/intro_nlp/ ? (y/n)y
```

ib > site-packages > pyhanlp > static >

名称

- __pycache__
- data
- __init__.py
- hanlp.properties
- hanlp.properties.in
- hanlp-1.8.3.jar
- index.html
- README

HanLP安装 1.x

```
(base) PS C:\Users\gerry_17578261252713> python
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pyhanlp
>>> from pyhanlp import *
>>> for term in HanLP.segment('你好，欢迎在Python中调用HanLP的API'):
...     print(f"{term.word} \t {term.nature}")

你好      v1
，         w
欢迎      v
在         p
Python    nx
中         f
调用      v
HanLP     nx
的         ude1
API       nx
>>>
```

HanLP安装 2.x

🕒 pip install hanlp_restful==0.0.23

轻量级RESTful API

仅数KB，适合敏捷开发、移动APP等场景。简单易用，无需GPU配环境，秒速安装。语料更多、模型更大、精度更高，**强烈推荐**。服务器GPU算力有限，匿名用户配额较少，[建议申请免费公益API秘钥 auth](#)。

Python

```
pip install hanlp_restful
```

创建客户端，填入服务器地址和秘钥：

```
from hanlp_restful import HanLPClient  
HanLP = HanLPClient('https://www.hanlp.com/api', auth=None, language='zh') # auth不填则匿名，zh中
```

```
(default) PS C:\Users\19410> pip install hanlp_restful==0.0.23  
Looking in indexes: https://mirrors.aliyun.com/pypi/simple  
Requirement already satisfied: hanlp_restful==0.0.23 in d:\anaconda3\envs\default\lib\site-packages (0.0.23)  
Requirement already satisfied: hanlp-common in d:\anaconda3\envs\default\lib\site-packages (from hanlp_restful==0.0.23)  
(0.0.20)  
Requirement already satisfied: phrasetree>=0.0.9 in d:\anaconda3\envs\default\lib\site-packages (from hanlp-common->hanlp_restful==0.0.23) (0.0.9)  
  
[notice] A new release of pip is available: 24.0 -> 24.3.1  
[notice] To update, run: python.exe -m pip install --upgrade pip
```

HanLP安装 2.x

🕒 pip install hanlp==2.1.0b57

🔗 海量级native API

依赖PyTorch、TensorFlow等深度学习技术，适合**专业**NLP工程师、研究者以及本地海量数据场景。要求Python 3.6至3.10，支持Windows，推荐*nix。可以在CPU上运行，推荐GPU/TPU。安装PyTorch版：

```
pip install hanlp
```

- HanLP每次发布都通过了Linux、macOS和Windows上Python3.6至3.10的[单元测试](#)，不存在安装问题。

HanLP发布的模型分为多任务和单任务两种，多任务速度快省显存，单任务精度高更灵活。

多任务模型

HanLP的工作流程为加载模型然后将其当作函数调用，例如下列联合多任务模型：

```
import hanlp
HanLP = hanlp.load(hanlp.pretrained.mtl.CLOSE_TOK_POS_NER_SRL_DEP_SDP_CON_ELECTRA_SMALL_ZH) # 世
HanLP(['2021年HanLPv2.1为生产环境带来次世代最先进的多语种NLP技术。', '阿婆主来到北京立方庭参观自然语义和
```

```
(default) PS C:\Users\19410> pip install hanlp==2.1.0b57
Looking in indexes: https://mirrors.aliyun.com/pypi/simple
Requirement already satisfied: hanlp==2.1.0b57 in d:\anaconda3\envs\default\lib\site-packages (2.1.0b57)
Requirement already satisfied: hanlp-common>=0.0.20 in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (0.0.20)
Requirement already satisfied: hanlp-downloader in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (0.0.25)
Requirement already satisfied: hanlp-trie>=0.0.4 in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (0.0.5)
Requirement already satisfied: pynvml in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (11.5.0)
Requirement already satisfied: sentencepiece>=0.1.91 in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (0.2.0)
Requirement already satisfied: termcolor in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (1.1.0)
Requirement already satisfied: toposort==1.5 in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (1.5)
Requirement already satisfied: torch>=1.6.0 in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (1.13.1)
Requirement already satisfied: transformers>=4.1.1 in d:\anaconda3\envs\default\lib\site-packages (from hanlp==2.1.0b57) (4.42.0)
Requirement already satisfied: phrasetree>=0.0.9 in d:\anaconda3\envs\default\lib\site-packages (from hanlp-common>=0.0.20->hanlp==2.1.0b57) (0.0.9)
Requirement already satisfied: typing-extensions in d:\anaconda3\envs\default\lib\site-packages (from torch>=1.6.0->hanl
```

NLP基础_词性标注

nr p n p n v y
小明 在 教室 把 苹果 吃 了

n p nr v y
苹果 被 小明 吃 了

NLP基础_词性标注

- ⌚ 词性是词汇的基本语法属性。词性标注是在给定句子中判定每个词的语法范畴，确定其词性并加以标注的过程。
- ⌚ 在中文词汇中，一个词语一般只有1到2个词性，并且其中一个词性的使用频率会远远大于另一个，所以词性标注最简单的方式是从语料库中统计每个词对应的高频词性，作为默认词性。
- ⌚ **当前主流的词性标注手段和分词一样，是将句子的词性标注当做一个序列标注问题来解决，比如：HMM、CRF等。**

NLP基础_命名实体识别



NLP基础_命名实体识别

⌚ 命名实体识别(**Named Entity Recognition, NER**), 是指识别文本中具有特定意义的实体, 主要包括人名、地名、机构名、专有名词、时间、货币等信息的提取识别。通常包括两部分:

- ⌚ 实体边界识别

- ⌚ 确定实体类别 (人名、地名、机构名等)

⌚ 常用的实现方式有:

- ⌚ 基于规则的命名实体识别

- ⌚ **基于序列标注统计的命名识别识别: CRF、HMM、LSTM、Transformer等**

NLP基础_命名实体识别

⌚ NER标注方法:

⌚ BIO标注法: I(Inside)、O(Outside)、B(Begin)

⌚ I-xxx: 在xxx类命名实体的内部(外开始外的所有位置);

⌚ O: 不属于实体;

⌚ B-xxx: 是xxx类命名实体的开始;

⌚ **BIOES标注法**: B(Begin)、I(Inside)、O(Outside)、E(End)、S(Single)

⌚ B-xxx: 是xxx类命名实体的开始;

⌚ I-xxx: 在xxx类命名实体的内部;

⌚ O: 不属于实体;

⌚ E-xxx: 在xxx类命名实体的结尾;

⌚ S-xxx: 单独属于xxx类命名实体。

NLP基础_关系抽取

- ⌚ 关系抽取是命名实体识别之后的具体应用，其应用主要分为两个方向：
 - ⌚ 关系抽取：从一个句子中判断两个entity是否有关系，一般是一个二分类问题，指定某种关系。
 - ⌚ 关系分类：一般是判断一个句子中两个entity是哪种关系(包含了是否有关系这个信息)，属于多分类问题。
 - ⌚ NOTE: 一般情况下，我们所说的关系抽取实际上就是**关系分类**。
- ⌚ 案例：
 - ⌚ 文本: 周星驰，1962年6月22日生于香港，祖籍浙江宁波
 - ⌚ 关系: 周星驰 --> 出生日期 --> 1962年6月22日、周星驰 --> 出生地 --> 香港、周星驰 --> 祖籍 --> 浙江宁波

⌚ 功能:

⌚ 中文分词

⌚ 词性标注

⌚ 命名实体识别

⌚ 知识图谱关系抽取

⌚ 关键词提取

⌚ 文本摘要

⌚ 新词发现

1. 词性标注说明

n	普通名词
nt	时间名词
nd	方位名词
nl	处所名词
nh	人名
nhf	姓
nhs	名
ns	地名
nn	族名
ni	机构名
nz	其他专名
v	动词
vd	趋向动词
vl	联系动词
vu	能愿动词
a	形容词
f	区别词
m	数词
q	量词
d	副词
r	代词
p	介词
c	连词
u	助词
e	叹词
o	拟声词
i	习用语
j	缩略语
h	前接成分
k	后接成分
g	语素字
x	非语素字
w	标点符号
ws	非汉字字符串
wu	其他未知的符号

2. 命名实体说明 (采用BIO标记方式)

B-PER、I-PER	人名
B-LOC、I-LOC	地名
B-ORG、I-ORG	机构名

jiagu安装

🕒 pip install jiagu==0.2.3

安装方式

pip安装

```
pip install -U jiagu
```



如果比较慢，可以使用清华的pip源： `pip install -U jiagu -i https://pypi.tuna.tsinghua.edu.cn/simple`

源码安装

```
git clone https://github.com/ownthink/Jiagu
cd Jiagu
python3 setup.py install
```



```
(default) PS C:\Users\19410> pip install jiagu==0.2.3
Looking in indexes: https://mirrors.aliyun.com/pypi/simple
Requirement already satisfied: jiagu==0.2.3 in d:\anaconda3\envs\default\lib\site-packages (0.2.3)

[notice] A new release of pip is available: 24.0 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(default) PS C:\Users\19410> python
Python 3.9.19 (main, Mar 21 2024, 17:21:27) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import jiagu
>>> jiagu
<module 'jiagu' from 'D:\\anaconda3\\envs\\default\\lib\\site-packages\\jiagu\\__init__.py'>
>>> |
```

jiagu_关系抽取

```
import jiagu
```

```
text = '姚明1980年9月12日出生于上海市徐汇区，祖籍江苏省苏州市吴江区震泽镇，前中国职业篮球运动员，司职中锋，现任中职联公司董事长兼总经理。'
```

```
knowledge = jiagu.knowledge(text)
```

```
print(knowledge)
```

```
>>> import jiagu
>>> text = '姚明1980年9月12日出生于上海市徐汇区，祖籍江苏省苏州市吴江区震泽镇，前中国职业篮球运动员，司职中锋，现任中职联公司董事长兼总经理。'
>>> knowledge = jiagu.knowledge(text)
>>> print(knowledge)
[['姚明', '出生日期', '1980年9月12日'], ['姚明', '出生地', '上海市徐汇区'], ['姚明', '祖籍', '江苏省苏州市吴江区震泽镇']]
>>>
>>> |
```

THANKS!