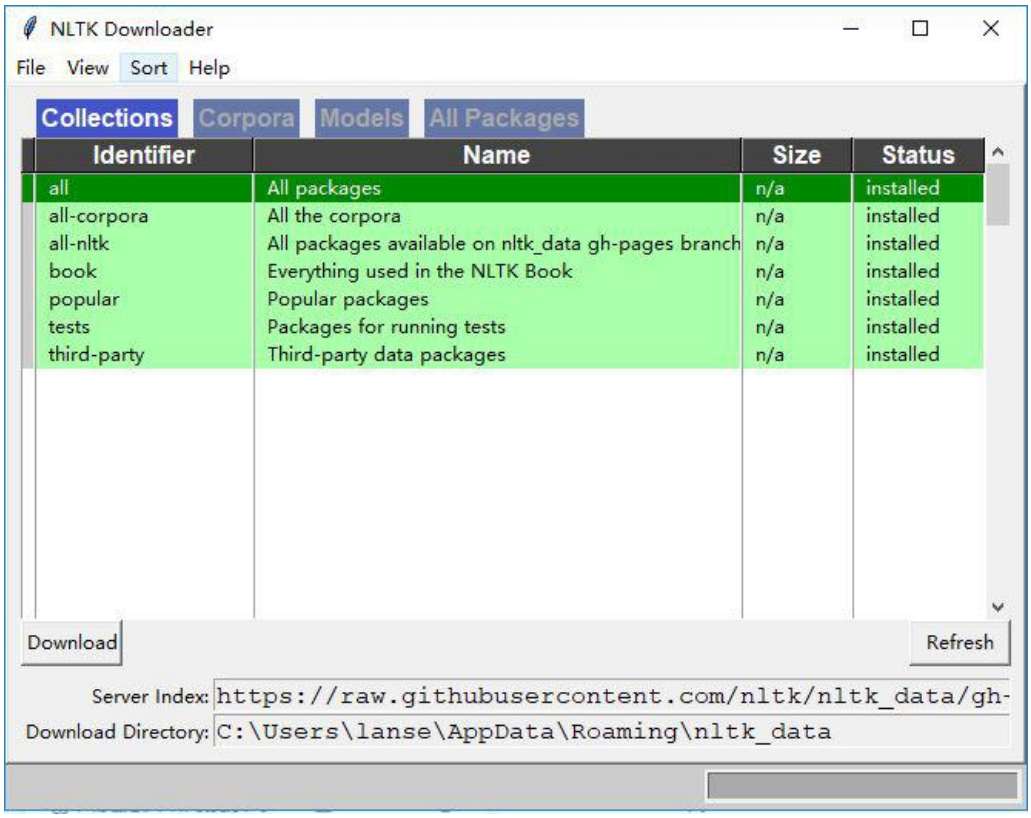
Python 自然语言处理

NLTK 入门(详细使用见官网：http://www.nltk.org/)

- 安装

```
pip install nltk
>>> import nltk
>>> nltk.download()
```



选择需要的包安装，建议默认路径下载，全部包安装大概需要 2G 内存

- 测试安装是否成功

```
>>> from nltk.book import *
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

NLTK 常见操作（英文语料）

- 文本切分成语句

```
import nltk
text="Don't hesitate to ask questions.Be positive."
from nltk.tokenize import sent_tokenize
print(sent_tokenize(text))

Out: ["Don't hesitate to ask questions.", 'Be positive.']
```

- 文本切分成语句（大批量句子切分、特定语言句子切分）

```
tokenizer=nltk.data.load('tokenizers/punkt/english.pickle')
print(tokenizer.tokenize(text))

Out: ["Don't hesitate to ask questions.", 'Be positive.']
```

- 分词方法 1：TreebankWordTokenizer 依据 Penn Treebank 语料库的约定，通过分离缩略词来实现切分

```
words=nltk.word_tokenize(text)
print(words)

Out: ['Do', "n't", 'hesitate', 'to', 'ask', 'questions', '.', 'Be', 'positive', '.']
```

- 分词方法 2：PunktWordTokenizer 通过分离标点来实现切分的，每一个单词都会被保留

```
from nltk.tokenize import WordPunctTokenizer
tokenizer=WordPunctTokenizer()
words = tokenizer.tokenize(text)
print(words)

Out: ['Don', "'", 't', 'hesitate', 'to', 'ask', 'questions', '.', 'Be', 'positive', '.']
```

- 其他分词方法 3：RegexpTokenizer、WhitespaceTokenizer、BlanklineTokenizer 等
- 频率分布 nltk.probability.FreqDist

fdist = FreqDist(samples)	创建包含给定样本的频率分布，参数为**词的列表**
fdist.inc(sample)	增加样本
fdist['monstrous']	计数给定样本出现的次数
fdist.freq('monstrous')	给定样本的频率
fdist.N()	样本总数
fdist.keys()	以频率递减顺序排序的样本链表
for sample in fdist:	以频率递减的顺序遍历样本
fdist.max()	数值最大的样本
fdist.tabulate()	绘制频率分布表
fdist.plot()	绘制频率分布图
fdist.plot(cumulative=True)	绘制累积频率分布图
fdist1 < fdist2	测试样本在 fdist1 中出现的频率是否小于 fdist2

- 条件频率分布 nltk.probability.ConditionalFreqDist

cfdist= ConditionalFreqDist(pairs)	从配对链表中创建条件频率分布
cfdist.conditions()	将条件按字母排序
cfdist[condition]	此条件下的频率分布
cfdist[condition][sample]	此条件下给定样本的频率
cfdist.tabulate()	为条件频率分布制表
cfdist.tabulate(samples, conditions)	指定样本和条件限制下制表
cfdist.plot()	为条件频率分布绘图
cfdist.plot(samples, conditions)	指定样本和条件限制下绘图
cfdist1 < cfdist2	测试样本在 cfdist1 中出现次数是否小于在 cfdist2 中出现次数

- nltk.text.Text()类用于对文本进行初级的统计与分析

Text(words)	对象构造,参数为词的列表
concordance(word, width, lines)	显示 word 出现的上下文
common_contexts(words)	显示 words 出现的相同模式
similar(word)	显示 word 的相似词
collocations(num, window_size)	显示最常见的二词搭配
count(word)	word 出现的词数
dispersion_plot(words)	绘制 words 中文档中出现的位置图
vocab()	返回文章去重的词典

- nltk.corpus 自带语料库

gutenberg	大约有 36000 本免费电子图书，多是古典作品
webtext	网络小说、论坛、网络广告等内容
nps_chat	有上万条聊天消息语料库，即时聊天消息为主
brown	一个百万词级别的英语电子语料库，这个语料库包含 500 个不同来源的文本，按文体分类有新闻、社论等
reuters	路透社语料库，上万篇新闻方档，约有 1 百万字，分 90 个主题，并分为训练集和测试集两组
inaugural	演讲语料库，几十个文本，都是总统演说

- 语料库操作

fileids()	返回语料库中文件名列表
fileids[categories]	返回指定类别的文件名列表
raw(fid=[c1,c2])	返回指定文件名的文本字符串
raw(catergories=[])	返回指定分类的原始文本
sents(fid=[c1,c2])	返回指定文件名的语句列表
sents(catergories=[c1,c2])	按分类返回语句列表
words(filename)	返回指定文件名的单词列表
words(catogories=[])	返回指定分类的单词列表

- 提取词干：词干提取可以被定义为一个通过去除单词中的词缀以获取词干的过程。以单词 raining 为例，词干提取器通过从 raining 中去除词缀来返回其词根或词干 rain。为了提高信息检索的准确性，搜索引擎大多会使用词干提取来获取词干并将其存储为索引词。
  - 方法 1：在 NLTK 中使用 PorterStemmer 类进行词干

```
import nltk
from nltk.stem import PorterStemmer
stemmerporter = PorterStemmer()
stemmerporter.stem('happiness')

Out: 'happi'
```

  - 方法 2：LancasterStemmer 类在 NLTK 中用于实现 Lancaster 词干提取算法

```
import nltk
from nltk.stem import LancasterStemmer
stemmerlan=LancasterStemmer()
stemmerlan.stem('happiness')

Out: 'happy'
```

  - 方法 3：在 NLTK 中，我们通过使用 RegexpStemmer 类也可以构建属于我们自己的词干提取器。它的工作原理是通过接收一个字符串，并在找到其匹配的单词时删除该单词的前缀或后缀

- 词性标注：词性标注是一个对句中的每个标识符分配词类（例如名词、动词、形容词等）标记的过程。在 NLTK 中，词性标注器存在于 nltk.tag 包中并被 TaggerIbase 类所继承

```
import nltk
text1=nltk.word_tokenize("It is a pleasant day today")
nltk.pos_tag(text1)

Out: [('It', 'PRP'), ('is', 'VBZ'), ('a', 'DT'), ('pleasant', 'JJ'), ('day', 'NN'), ('today', 'NN')]
```

一些文本处理操作

- 消除标点符号（中英文）

```
def filter_punctuation(words):
    new_words = [];
    illegal_char = string.punctuation + '【·！…（）—：“”？《》、；】'
    pattern=re.compile('[%s]' % re.escape(illegal_char))
    for word in words:
        new_word = pattern.sub(u'', word)
        if not new_word == u'':
            new_words.append(new_word)
    return new_words

words_no_punc = filter_punctuation(words)
print(words_no_punc)

Out: ['Don', 't', 'hesitate', 'to', 'ask', 'questions', 'Be', 'positive']
```

- 文本的大小写转换

```
print(text.lower())
print(text.upper())
Out:
don't hesitate to ask questions. be positive.
DON'T HESITATE TO ASK QUESTIONS. BE POSITIVE.
```

- 处理停止词（英文）

```
from nltk.corpus import stopwords
stops=set(stopwords.words('english'))
words = [word for word in words if word.lower() not in stops]
print(words)
```

中文语料处理

- 查看 jieba 分词的使用：https://github.com/fxsjy/jieba
- 处理获得分词列表进一步处理