计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(1):275−278 [doi: 10.15888/j.cnki.csa.006700]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 NLTK 的中文文本内容抽取方法①

李　晨, 刘卫国

(中南大学 信息科学与工程学院, 长沙 410083)
通讯作者: 刘卫国, E-mail: liuwg@csu.edu.cn

摘　要: NLTK 是 Python 中用于自然语言处理的第三方模块, 但处理中文文本具有一定局限性. 利用 NLTK 对中文文本中的信息内容进行抽取与挖掘, 采用同语境词提取、双连词搭配提取、概率统计以及篇章分析等方法, 得到一个适用于中文文本的 NLTK 文本内容抽取框架, 及其具体的实现方法. 经实证分析表明, 在抽取结果中可以找到反映文本特点的语料内容, 得到抽取结果与文本主题具有较强相关性的结论.

关键词: 自然语言处理; 中文文本; 自然语言处理工具包

引用格式: 李晨,刘卫国.基于 NLTK 的中文文本内容抽取方法.计算机系统应用,2019,28(1):275–278. http://www.c-s-a.org.cn/1003-3254/6700.html

Chinese Text Information Extraction Based on NLTK

LI Chen, LIU Wei-Guo

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: NLTK is a module for processing natural language text in Python, but it has limitations when processing Chinese text. To extracted information in the text by using NLTK, the means created in this study included a group of methods, such as common context words extraction, bigrams words extraction, probability statistics, and discourse analysis. Both of NLTK text content extraction framework suitable for Chinese texts and implementation method are obtained. In the results of empirical, it finds the content of the corpus which reflects the characteristics of the text, and gets the conclusion that a strong correlation between the results of extraction and text topic.

Key words: natural language processing; Chinese texts; NLTK

NLTK 的默认处理对象是英文文本, 处理中文文本存在一定的局限性, 主要体现在以下两点:

(1) NLTK 素材语料库中缺少中文语料库, 在 NLTK 模块中包含数十种完整的语料库, 例如布朗语料库、古腾堡语料库等, 但没有中文语料库. 另外, NLTK 也没有中文停用词语料库, 在文本预处理中, 特别是在进行频率统计之前需要使用停用词语料库对文本进行过滤清洗, NLTK 没有提供针对中文的停用词库, 使用针对英文的过滤方法是无法完成中文文本的停词过滤.

(2) 在英文文本中, 文本的分割可以由单词之间的空格完成, 但是中文文本的分割依靠 NLTK 是无法完成的, 中文分词工作需要借助分词工具来完成, 已有的一些中文分词工具有结巴分词 (jieba)、斯坦福中文分词器等.

本文应用 NLTK 对中文文本进行信息抽取.

1　NLTK 文本内容抽取框架

使用 NLTK 对自然语言文本中词句内容进行提取与分析, 可以概括为筛选、提取、统计、解释的过程[1,2]. 抽取方法首先对其中无实际含义的词汇进行筛选过滤, 使用概率统计方法提取出高频词集, 接着识别出文本的关键词, 选定研究词汇. 以该词汇作为目标词,

① 收稿时间: 2018-05-28; 修改时间: 2018-06-19; 采用时间: 2018-07-05; csa 在线出版时间: 2018-12-26

查找其所在的句子, 对段落、篇章内的词汇进行计数统计与篇章分析, 解释文本内容与研究问题. 为此将以上过程简化为三个阶段, 分别是预处理、分析以及输出阶段, 如图 1 所示.

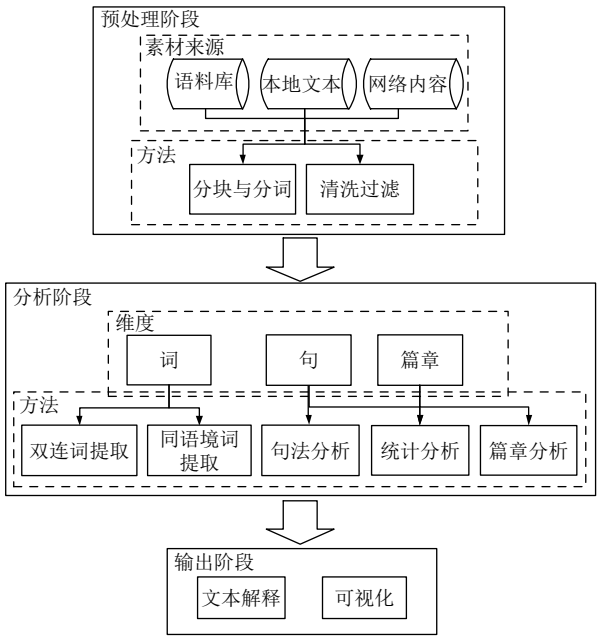

图 1 NLTK 抽取框架

1.1 预处理阶段

预处理阶段是确定文本并对文本进行简单处理的过程, 预处理方法主要有分块与分词、清洗过滤. 分块与分词是对文本进行切分的操作, 由于分析阶段各方法调用对象的不同, 文本需要划分成不同子单位的文本, 例如以词为最小单位的文本、以句为最小单位的文本. 清洗过滤是对文本分词处理后删去文本中无实际意义的词汇符号的过程.

1.2 分析阶段

分析阶段是抽取过程的核心, 涉及到文本的一系列处理操作. 文本经预处理阶段以词、句、篇章为基本单位进行切分后, 在这三个维度上使用不同的分析方法. 经分词处理后, 针对单个词汇进行的操作有双连词提取、同语境词提取. 在单个句子上可执行的操作有词性标注、句法分析. 在篇章中可以进行篇章分析[3]与统计分析, 其中统计分析是最常使用的工具. 例如, 在对布朗语料库[4]的研究中, 以概率统计的方法得到不同文体中情态动词的频率分布, 归纳总结出情态动词在文体中的分布规律, 从而对文本的文体进行判断.

1.3 输出阶段

文本的抽取结果分为两大类, 一类是将提取出词与句子等结果作为原始依据, 对文本内容含义进行解释; 第二类是以可视化的形式展示数据结果, 更加直观地体现词汇频率的对比.

2 NLTK 对文本的处理方法

2.1 预处理方法

2.1.1 对原始文本分词与分句

在 NLTK 中, 分词分句操作可以将文本处理成可以单独调用的词或句子. 分词是将句子序列或字符串构成的文本划分成单个的词. 分句是将文本中的篇章段落划分成单个的句子. 使用 nltk.word_tokenize()、nltk.sent_tokenize() 进行分词与分句操作. 分词处理后文本中的词汇、符号转化为单一标识符, 这是进行后续分析工作的关键.

2.1.2 对原始文本进行清洗过滤

清洗与过滤实际上是一个分类的过程, 使用正则表达式匹配需要过滤掉的数字与符号, 对文档中的纯文本内容进行提取. 然后, 利用停用词语料库对文本实现过滤, 停用词语料库中包含无实际意义的高频词汇, 例如 a, to 等.

以下的命令定义了一个过滤英文停用词的函数, 将文本中的词汇归一化处理为小写并提取, 从停用词语料库中提取出英语停用词, 使用词汇运算符将文本进行区分.

```
def uwords(text):
    text_vocab=[w.lower() for w in text if w.isalpha()]
    stopwords=set(nltk.corpus.stopwords.words('english'))
    uwords=[w for w in text_vocab if w not in stopwords]
```

2.2 分析方法

2.2.1 同语境词查找

NLTK 中使用函数 similar() 查找与目标词汇出现在相似上下文位置的词, 即在文本中可用作替换的词汇. 在《白鲸记》中使用 text.similar("captain") 找到以下同语境词: whale ship sea boat deck world other. 可以发现得到的词汇与目标词汇词性均为名词.

2.2.2 统计分析

概率统计作为最常用的数学分析手段, 用于文本中数据的处理分析. 在 Python 中借助 NLTK 频率分布类中的函数, 对文本中出现的单词、搭配、常用表达、符号进行频率统计、长度计算等相关操作, 使用

fdist= nltk.FreqDist() 对研究文本创建频率分布, 函数 fdist['target word']查找频率分布内目标词汇的出现次数, fdist. most_common(*n*) 从频率分布中提取出高频词汇, 其中参数 *n* 为提取词汇的数量.

2.2.3 篇章分析

对篇章内容的分析也是使用 NLTK 对文本内容进行抽取的方法之一. 布朗语料库以文体作为分类标准. 根据这一特点, 利用布朗语料库探索词汇在不同文体中的使用情况. 使用条件概率的方法, 选择布朗语料库中 6 个不同的文体类型分别统计 wh-词的使用情况. 命令如下:

```
cfd=nltk.ConditionalFreqDist(
    (genre, word)
    for genre in brown.categories()
    for word in brown.words(categories=genre))
cfd.tabulate(conditions=genres, samples=modals)
```

命令执行后, 以表格的形式打印所得结果如图 2 所示.

	what	when	where	which	who	why
news	76	128	58	244	268	9
religion	64	53	20	202	100	14
hobbies	78	119	72	252	103	10
science_fiction	27	21	10	32	13	4
romance	121	126	54	104	89	34
humor	36	52	15	62	48	9

图 2 频率统计结果

从结果上来看, 在 religion 与 news 文体中 which 和 who 出现次数较多, hobbies 文体中 when、which、who 出现较多, 而在 science_fiction 与 humor 文体中 wh-词出现的次数较为均匀, 在 romance 文体中 what、when、which 出现次数较多. 使用条件概率的方法找到文体中同类型词使用差异, 以表格的形式展示结果[5].

抽取过程中叠加使用分析方法, 将获得的结果作为分析的原始素材进行二次加工处理.

2.3 输出方法

利用 Python 的第三方图表模块对数据进行二次处理. Matplotlib 是 Python 中用于可视化处理的模块, 该模块对结构化数据绘制统计图表, 例如柱状图、扇形图、折线图、直方图等图形[6]. 导入 Matploylib 库, 并引入上一阶段获得的数据. 函数 fdist.plot(*n*, cumulative=True/False) 在建立概率分布的基础上对统计结果绘制频率折线图, 其中参数 *n* 表示折线图展示的词汇数, cumulative 表示是否对统计结果逐词累加. 经函数 text.dispersion_plot() 处理后获得离散图, 横坐标为文本词汇排序, 纵坐标为研究词汇, 词汇对应的每一行代表整个文本, 一行中的一条竖线表示一个单词, 以竖线的排列表示词汇在文本中的位置.

3 实证分析

以 2018 年政府工作报告为案例素材进行分析. 在网络上通过爬虫工具得到 2018 年政府工作报告, 总字数为 20 257. 根据本文提出的方法步骤, 使用 NLTK 对文本内容进行抽取.

3.1 方法描述

对目标素材所在网页进行内容提取. 使用 Scrapy 获取网页上的报告内容, Scrapy 是用于获取网页内容的 Python 第三方模块, 多用于爬取网页上的图片以及结构化的文本内容. 将爬取得到的文本内容保存在 txt 文件中, 过滤文本中的标点符号、数字与停用词等. 以目标词汇为中心进行二次分析, 使用到同语境词查找, 双连词查找等方法. 使用概率统计对高频出现的词汇进行查找输出, 最后以折线图、离散图等可视化的形式展示结果. 实例分析流程如图 3 所示.

在预处理阶段, 使用正则表达式清理爬取内容中的符号、数字以及英文字母, 保留中文文本. join(re.findall (r'[\u4e00-\u9fa5]', raw_text)). 采用 Jieba 中文分词工具对中文文本分词 jieba.lcut().

分析阶段查找 “民生” 所在的句子以及查找与 “发展” 位于相同位置的词汇:

```
text.concordance(word='民生')
text.similar('发展').
```

3.2 结果展示

应用 matplotlib 对结果进行可视化, 输出离散图展示不同高频词在文本中的位置:

```
text.dispersion_plot(['经济','企业','工作','社会'])
```

输出离散图展示“经济”, “企业”, “工作”, “社会” 四个词在文本中的位置, 结果如图 4 所示.

对去除停用词后的文本, 进行频率统计. 将前 20 个高频词以折线图的形式输出, 结果如图 5 所示.

在中文文本的分析中, 出现可视化结果无法显示中文的问题, 对此解决的方法是增加部分代码, 指定默认中文字体. 命令如下.

```
import matplotlib as mpl
mpl.rcParams['font.sans-serif'] =['SimHei']
```

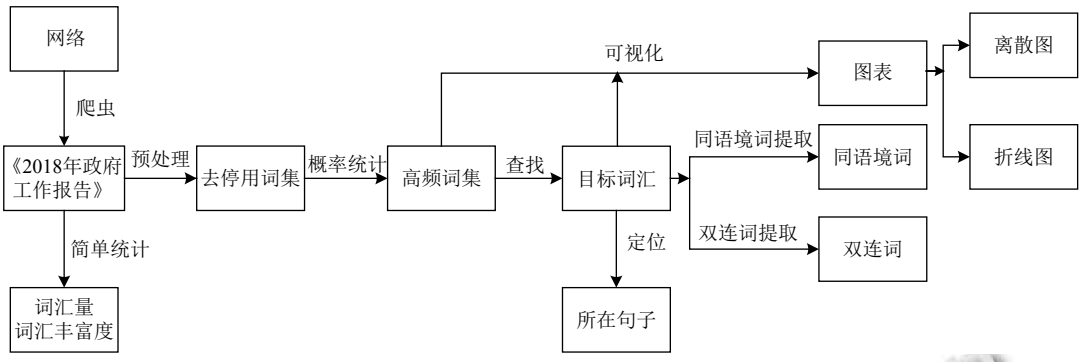

图 3 实例验证流程图

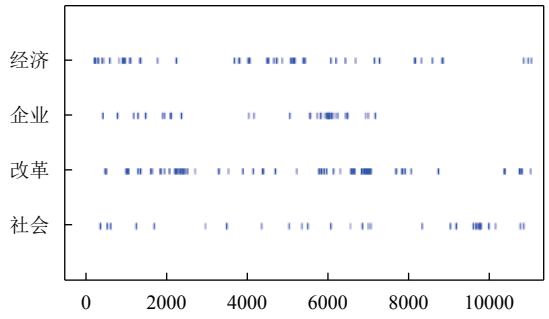

图 4 展示不同词在文本中的位置

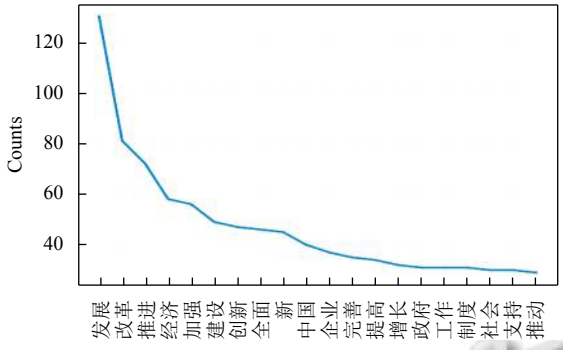

图 5 前 20 个高频词折线图

3.3 分析与启示

从概率统计的结果中, 发现提取出的词汇具有两个特征: 集中在名词与形容词; 词汇含义与文本主题相关. 从离散图 (图 4) 来看, “经济” 一词的在全文中不同位置均有出现, 且在前一部分相对集中分布; “社会”一词在报告的全文中均匀分布, 可以看出关注社会、关注经济体现在整篇工作报告当中. 对比 “改革”与 “社会”, 可以发现“改革”的出现频度超过“社会”, 也印证了图 5 的频率统计结果. 在高频折线图中 (图 5) 排在前两位的词是“发展” 以及“改革”, 这与“进一步深化改革”保持一致; 在前十个高频词中“经济”作为出现次数最多的名词, 也说明了经济的重要性. 在报告抽取结果中找到了反映政府工作报告的语料内容, 达到了理解语料库的目的.

4 结束语

由于中英文文本有不同的分词方法, 使得 NLTK 在中文文本处理上存在不足, 中文文本不能通过简单的字符分隔来达到语义分隔的目的, 需要由分词工具来完成, 并且可能存在歧义.

本文加入爬虫工具与 NLTK 协同作用, 对爬虫得到的文本使用分词工具、正则表达式完成对中文文本中词汇与句子等内容的预处理工作, 并在 NLTK 中完成对中文文本的统计与分析, 抽取出与主题相关的文本内容, 实现 NLTK 对中文文本上的处理, 达到对中文文本信息抽取的目的.

参考文献

1 化柏林, 张新民. 从知识抽取相关概念辨析看知识抽取的特点和发展趋势. 情报科学, 2010, 28(2): 311–315.
2 丁玉飞, 王曰芬, 刘卫江. 面向半结构化文本的知识抽取研究. 情报理论与实践, 2015, 38(3): 101–106.
3 李保利, 陈玉忠, 俞士汶. 信息抽取研究综述. 计算机工程与应用, 2003, (10): 1–5, 66. [doi: 10.3321/j.issn:1002-8331.2003.10.001]
4 Bird S, Klein E, Loper E. Natural Language Processing with Python. California: O’Reilly Media, 2009. 42–44.
5 邓擎琼, 彭炜明, 尹乾, 等. Python 教学中实用型词频统计案例展示. 计算机教育, 2017, (12): 20–27. [doi: 10.3969/j.issn.1672-5913.2017.12.005]
6 张莉, 金莹, 张洁. 基于 MOOC 的 “用 Python 玩转数据” 翻转课堂实践与研究. 工业和信息化教育, 2017, (3): 70–76. [doi: 10.3969/j.issn.2095-5065.2017.03.015]