

关键词提取

1. 什么是关键词

《现代汉语词典》中对关键词的解释有两个义项：一是“指能体现一篇文章或一部著作的中心概念的词语”；二是“指检索资料时所查内容中必须有的词语”。在正式的学术论文中都要求标引“关键词”一项，它通过几个词语的逻辑组合表达文章的主题，同时被应用于信息检索。学术论文中的关键词显然是《现汉》说的第一个意思，学术论文中的关键词一般是作者写论文时自行总结的。

1987 年颁布的国家标准《科学技术报告、学位论文和学术论文的编写格式》（GB7713-87）有一个更为详细的描述：“关键词是为了文献标引工作，从报告、论文中选取出来用以表示全文主题内容信息款目的单词或术语。”

2. 为什么需要提取关键词

在现代社会，为了要搜索到需要的信息，我们需要从海量的文本中将相关的文本搜集到一起进行进一步的研究。如何判断一篇文本跟我们的研究兴趣是否相关呢？通过查看文本的关键词无疑是一条最方便的途径。论文作者写论文的时候已经自行提供了关键词，但网上大量的文本并没有人为标注的关键词，对于这种没有标注关键词的文本，我们可以用某种算法自动将文本的关键词提取出来，然后利用提取结果再来判断该文本跟我们的研究兴趣是否相关。

显然，文本关键词自动提取的准确程度直接关系到后续研究的进度和结果。准确的关键词提取结果有助于我们收集准确的数据，不准确的关键词将会导致我们得到跟研究目的不相关的数据，从而延缓和阻碍研究的进一步开展。

所以，有必要研究关键词自动提取的算法。

3. 最简单的关键词提取思路

什么样的词才是关键词呢？能反映文本主题的词。反映文本主题的关键词显然会在文本中反复出现。所以，一个直接的想法就是：如果一个词在文本中出现的次数越多（也就是频次越高），该词是关键词的可能性就越大。为此，我们可

以对要提取关键词文本搞一个简单的词频统计。

当然，搞词频统计之前需要用分词软件将文本切分。可以借助于现有的分词软件，也可以自己写一个程序。总之，假设我们现在有了切分后的文本。对于切分后的文本，如何统计词频，也是很简单的，会写程序就自己写一个词频统计程序，不会写程序就找一个词频统计软件好了，找这种软件当然难不倒我们。

好了，我找到了 7 个文本：《明朝那些事》的七册书的做电子版，提前做好了分词。7 个文本的信息如下（文本编号、文件名、字节数）：

| | | |
|------|--------------|-----------|
| 文本 1 | 明朝那些事儿·壹.txt | 985,101 |
| 文本 2 | 明朝那些事儿·贰.txt | 945,094 |
| 文本 3 | 明朝那些事儿·叁.txt | 777,782 |
| 文本 4 | 明朝那些事儿·肆.txt | 908,241 |
| 文本 5 | 明朝那些事儿·伍.txt | 1,003,419 |
| 文本 6 | 明朝那些事儿·陆.txt | 893,625 |
| 文本 7 | 明朝那些事儿·柒.txt | 861,576 |

现在想求文本 1 的关键词，先看词频统计结果的前 10 名，同时，作为对比，也给出全部 7 个文本的词频统计结果的前 10 名：

| 文本 1 | | | 所有文本 | | |
|------|----|-------|------|----|--------|
| 序号 | 词语 | 频次 | 序号 | 词语 | 频次 |
| 1 | , | 15158 | 1 | , | 113929 |
| 2 | 的 | 9297 | 2 | 的 | 48815 |
| 3 | 。 | 4426 | 3 | 。 | 31013 |
| 4 | 了 | 3406 | 4 | 了 | 24145 |
| 5 | 他 | 3381 | 5 | 是 | 20263 |
| 6 | 是 | 3283 | 6 | 他 | 19417 |
| 7 | 一 | 1923 | 7 | 不 | 12980 |
| 8 | 不 | 1913 | 8 | 一 | 12380 |
| 9 | 在 | 1680 | 9 | 人 | 10397 |
| 10 | 人 | 1486 | 10 | 这 | 9512 |

我们发现，文本 1 的前 10 名都是一些常用的词，看了它们根本不知道文本 1 的主题。而且，文本 1 的前 10 名中有 9 个词同时也进入了所有文本的词频排行榜的前 10 名榜单。

看来，纯粹靠词频统计来发现关键词，还真不靠谱。原因就在于有很多常用词在一个文本中是高频词，在其它文本行中也有可能是高频词，所以用这些词来作为文本的关键词显然是不合适的另外，标点符号这样的词显然也不适合充当关

关键词，所以标点符号也得去掉。

有些人就想了这么一个主意：建立一个词表，凡是这样的对区分文本主题没有作用的词都放入词表，在统计词频的时候，凡是该词表中收录的词直接忽略掉，不予统计其词频。这样的词表被称为“停用词表”。停用词主要包括字母、数字、标点符号及使用频率特高的单字词和个别高频率的双字词等。停用词表也可以只包含纯粹汉字词，包含字母、数字和标点符号的词可以在程序代码中用正则表达式过滤掉不予统计，这样也减轻了停用词表的维护负担。

假定我们下载的停用词表叫“汉语停用词表.txt”，自己用 Python 写了一个词频统计程序，在词频统计时使用停用词表，并且只统计纯汉字词的频次。现在再来看前 10 名的词频分布结果。

| 文本 1 | | | 所有文本 | | |
|------|-----|------|------|----|------|
| 序号 | 词语 | 频次 | 序号 | 词语 | 频次 |
| 1 | 朱元璋 | 1162 | 1 | 年 | 3258 |
| 2 | 朱棣 | 725 | 2 | 说 | 3053 |
| 3 | 中 | 480 | 3 | 中 | 2272 |
| 4 | 说 | 388 | 4 | 皇帝 | 2121 |
| 5 | 军 | 379 | 5 | 位 | 2028 |
| 6 | 年 | 325 | 6 | 死 | 1734 |
| 7 | 陈友谅 | 323 | 7 | 两 | 1557 |
| 8 | 元 | 282 | 8 | 天 | 1450 |
| 9 | 时 | 228 | 9 | 想 | 1425 |
| 10 | 想 | 218 | 10 | 朱棣 | 1346 |

根据上表的数据，我们可以看到，文本 1 里面的高频词前 10 名就有一些词可以作为关键词了，比如“朱元璋”“朱棣”“陈友谅”。像“中”“说”之类的高频词还不适合做关键词，因为他们在全部文本中出现的频次也相当高，并不能算是文本 1 特有的词，并不能反映文本 1 的主题思想。要想过滤这些词，加大停用词表即可。

可见，停用词表还是有一定的作用的。但是停用词表很难适用不同的领域，维护比较麻烦。

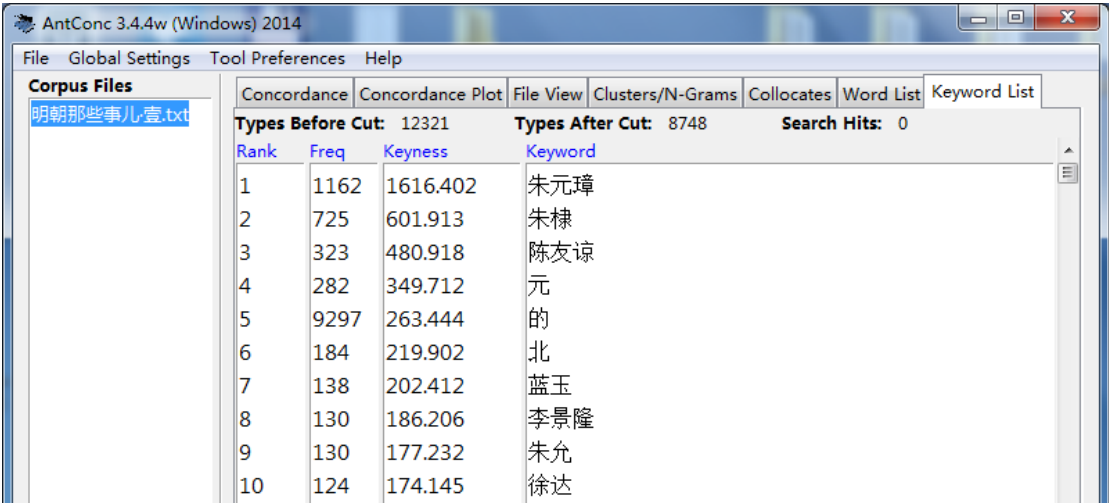
4. 稍微复杂一点的关键词提取思路

某词的重要性与它在目标文献中的词频成正比，而与该词在参考语料库中的词频成反比。如果某一词 A 在目标文献中为高频词，但在参考语料库中出现的频

率低，则 A 就被认为是目标文献的关键词。这种方法可以避免类似“的”“是”这样的高频率词出现在每一篇文献的关键词表里面。

AntConc 可以做这个事情。在生成某一具体文本的关键词表的时候，需要一个已分好词的参考语料库（reference corpus），然后先生成包含目标文本的所有语料的词表，再对比目标文本的小词表中词语的词频和参考语料库中同一词的词频，并根据对比结果计算出关键度（keyness）。如果某一个词在小词表中的词频比它在参考语料库中的词频高到一定程度，那么这一个词就是目标文件的关键词之一，高出越多，关键度越高。

下面是利用全部 7 篇文本当参考语料库对文本 1 进行关键词提取的结果：



| AntConc 3.4.4w (Windows) 2014 | | | |
|---|------|-----------------------|----------------|
| File Global Settings Tool Preferences Help | | | |
| Corpus Files | | | |
| 明朝那些事儿.壹.txt | | | |
| Concordance Concordance Plot File View Clusters/N-Grams Collocates Word List Keyword List | | | |
| Types Before Cut: 12321 | | Types After Cut: 8748 | Search Hits: 0 |
| Rank | Freq | Keyness | Keyword |
| 1 | 1162 | 1616.402 | 朱元璋 |
| 2 | 725 | 601.913 | 朱棣 |
| 3 | 323 | 480.918 | 陈友谅 |
| 4 | 282 | 349.712 | 元 |
| 5 | 9297 | 263.444 | 的 |
| 6 | 184 | 219.902 | 北 |
| 7 | 138 | 202.412 | 蓝玉 |
| 8 | 130 | 186.206 | 李景隆 |
| 9 | 130 | 177.232 | 朱允 |
| 10 | 124 | 174.145 | 徐达 |

我们从上图可以看到，“朱元璋”“朱棣”“陈友谅”这三个词的关键度最高，跟前面用停用词表得到的结果是吻合的。

这种做法放弃了停用词表，有一定的先进性，但它利用了现有的关键词提取软件，灵活性太差。如果参考语料库有一万篇文本，要求出每一篇文本的关键词，我们就得手工操作一万次，这工作量，吓死个人啊。所以，是时候发挥程序猿的优势了，写代码来实现关键词的提取。

5. 流行的关键词提取思路

比较简单一点的方法是使用 TF-IDF 方法。

TF-IDF 是提取关键词的最基本、最简单易懂的方法。判断一个词在一篇文章中是否重要，一个容易想到的衡量指标就是词频，重要的词往往会在文章中多次出现。但另一方面，不是出现次数多的词就一定重要，因为有些词在各种文章中都频繁出现，那它的重要性肯定不如那些只在某篇文章中频繁出现的词重要性

强。从统计学的角度，就是给予那些不常见的词以较大的权重，而减少常见词的权重。IDF（逆文档频率）就是这个权重，TF 则指的是词频。

TF = （词语在文本中出现次数） / （文章总词数）

IDF = \log （语料库文本总数 / （包含该词的文档数+1））

TF-IDF = TF * IDF

这些数据都是很容易计算的，NLTK 就可以直接用这个办法提取关键词。

6. 更复杂的关键词提取思路

清华大学刘知远老师的博士论文《基于文档主题结构的关键词抽取方法研究》就是专门研究关键词提取的。论文下载地址：

https://link.zhihu.com/?target=http%3A//nlp.csai.tsinghua.edu.cn/%7Elzy/publications/phd_thesis.pdf

这里面详细描述了关键词提取的技术。