计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(7):162−166 [doi: 10.15888/j.cnki.csa.006450]

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

多特征关键词提取算法研究①

王　洁, 王丽清

(云南大学 信息学院, 昆明 650223)
通讯作者: 王丽清, E-mail: wlq@ynu.edu.cn

摘　要: 关键词提取技术是语料库构建、文本分析处理、信息检索的基础. 采用传统的 TFIDF 算法提取关键词时, 主要依据词频计算权重, 而未考虑文本特征项的影响, 由于对词频的过度依赖, 导致其提取关键词的准确性不高. 针对这个问题, 本文根据关键词的特性, 引入位置和词性作为影响因子, 对 TFIDF 算法权重重新进行了计算和排序, 从而改进该算法, 并利用 Python 语言完成了实现. 实验结果表明, 采用该改进方法提取关键词, 其召回率、准确率、*F* 因子与传统方法相比均得到明显提升.

关键词: 多特征; TFIDF; 关键词提取; Python

引用格式: 王洁,王丽清.多特征关键词提取算法研究.计算机系统应用,2018,27(7):162–166. http://www.c-s-a.org.cn/1003-3254/6450.html

Research on Multi-Feature Keyword Extraction Algorithm

WANG Jie, WANG Li-Qing

(School of Information Science &Engineering, Yunnan University, Kunming 650223, China)

Abstract: Keyword extraction technology is the foundation of corpus construction, text analysis, and information retrieval. The traditional TFIDF algorithm is mainly based on word frequency weighting to extract keywords without considering the influence of text features. The excessive reliance on word frequency leads to the inaccuracy of extract keywords. To solve this problem, an improved algorithm has been proposed, which use the word position and the word information as factors to recalculate the weight, then we implement it in Python. Experiment shows that using this method to extract keywords can improve the recall rate, accuracy, and *F*-measure.

Key words: multi-feature; TFIDF; keyword extraction; Python

随着互联网的飞速发展, 网络资源的数据量日益庞大. 面对海量的数据, 如何有效、准确地提高对文章内容的检索效率, 一直是一个研究热点. 一篇文章的关键词往往反映了文章的主题, 具有与内容高度相关的代表性. 准确、快速地提取关键词, 对于信息检索至关重要.

近年来, 国内外相关学者在关键词提取技术领域开展了大量的研究工作. 其中, TFIDF (Term Frequency-Inverse Document Frequency) 算法主要通过对词频的计算来判断提取词对文章的代表性, 是常用的一种基于统计的提取方法. 但却存在过度依赖词频的问题. 为此, 往往会引入特定因子进行计算以此减少对词频的依赖. 例如, 考虑基于词位置和词跨度的方法进行 TFIDF 算法的权重改进[1,2], 或者利用语义的连贯性, 结合词频和位置特征, 进行加权[3], 也有引入信息熵的方法[4]. 但这些方法存在一定计算复杂性问题, 或者在文

① 基金项目: 云南省教育厅产业化扶持项目 (2016CYH03); 云南省科技创新强省计划项目 (2014AB021); 云南省创新团队项目
Foundation item: Industrial Support Project of Education Department of Yunnan Province (2016CYH03); Science and Technology Innovation and Strong Province Plan Project of Yunnan Province (2014AB021); Innovation Team Project of Yunnan Province
收稿时间: 2017-11-23; 修改时间: 2017-12-15; 采用时间: 2017-12-27; csa 在线出版时间: 2018-06-27

章类型和语料规模上有一定局限. 也有结合文章综合信息和引用新闻类别因子的方法[5,6], 增加了其他特征因子对权重进行加权, 能一定程度校正词频依赖问题. 但未考虑关键词词性以及关键词覆盖度不同所带来的影响.

针对上述问题, 本文将 TFIDF 算法和中文语言的特性进行了有效结合, 通过引入位置因子、词性因子等多特征值进行权重的二次计算, 校正词频依赖造成的关键词偏差, 并利用 Python 完成算法实现验证. 实验结果表明词性因子和位置因子的引入, 计算简单、高效, 能够有效地提高中文关键词的提取效果, 适于短文章关键词的提取.

1 基于 TFIDF 的多特征关键词提取改进算法

1.1 TFIDF 算法

TFIDF 算法基本思想是: 利用词频 (Term Frequency, TF) 和逆文档频率 (Inverse Document Frequency, IDF) 相乘, 得到权重值. 权重越大, 则该词为关键词的概率越高[7]. 根据 TFIDF 算法, 词权重 $W_{tf}(i)$ 的计算公式如下:

$$W_{tf}(i) = tf_i * idf_i \quad (1)$$

式 (1) 中, tf_i=该词在文章内容中出现的次数/文章总词数, tf_i 值表示该词在文档中出现的频率; idf_i=log(语料库文档总数/包含该词的文档数), 即: idf_i=log(N/df_i), idf_i 值的大小代表了该词的类别区分能力[8].

根据以上公式可知, 当词在文档中出现频率较高, 而在包含该词的文档数中出现频率较低时, 则该词根据 TFIDF 算法得到的权重值 W_{tf} 就越高, 故该词可以在一定程度上表示文章的内容.

1.2 算法改进思想

(1) 特征分析

位置因子. 分析新闻或信息内容的特征可知: 新闻篇幅相对较短, 其标题往往具有概括性. 如果一个词, 在标题中出现, 该词的重要性往往高于其他词. 词语出现的位置在一定程度上反映了词语的重要性[9].

词性因子. 词性是一种浅层语言学知识的表示, 根据汉语词性可分为实词和虚词. 实词包含: 名词、动词、形容词、数词、量词和代词. 虚词包括: 副词、介词、连词、助词、叹词、拟声词. 关键词词性分布一般是名词或名词性短语为主. 其次是动词, 最后是数词、副词和其他修饰词等[5]. 考虑词性特征可以有效避免传统采用语言学方法的缺陷[10].

(2) 加权方式

TFIDF 算法在提取中, 主要依赖词频进行权重计算, 但存在过度依赖词频的噪声. 因此, 可以先利用 TFIDF 算法完成权重计算后, 再对每个词进行位置判断. 并且, 在考虑标题位置的同时也考虑该词在正文中的位置. 将同时出现在标题和正文中的词, 其因子设置为最高; 只出现在标题中的次之; 而只出现在正文中的因子设为逐次递减的不为零的低值.

完成位置特征引入后, 再进一步考虑引入词性特征. 对提取出来的关键词进行词性标注. 按名词、动词和其他词汇的顺序从高到低设置为不为零的值.

最后, 将以上词频、标题位置、正文位置和词性作为多特征因子, 共同引入权重计算中, 计算出每个特征词的综合权重, 按其排序进而确定关键词. 这就是本文算法的核心思想.

1.3 关键词提取流程

关键词的提取流程示意图如图 1.

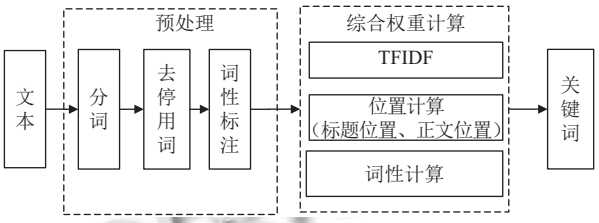

图 1 关键词提取

在关键词提取前, 首先需要对文本进行分词和去停用词等预处理. 分词是进行中文关键词提取的第一步, 也是重要的一步. 通过分词将文本中的每一个句子, 按照一定的规则划分成有序的词语片段, 其中标点、词语、单字均可以划分. 本文利用 Python 语言结合结巴分词进行实现.

停用词在文本分析中, 属于一种冗余数据, 对文章的主题不具备表达能力, 往往具有高频、无意义等特点. 例如: “的”、“啊”、“但是”等词语以及标点符号. 通过去除停用词, 能消除对关键词提取的干扰. 在本文中, 采用引入对应的停用词表并保存, 再以 Python 编程调用完成去除处理.

在 Python 中, 词性的标注规则如表 1 所示, 名词均是以 n 开头, 动词以 v 开头.

表 1 部分结巴分词词性标注

代码	名称
A	形容词
Ad	副形词
an	名形词
ag	形容词性语素
al	形容词性惯用语
n	名词
nr	人名
ns	地名
nt	机构团体名
nz	其它专名
nl	名词性惯用语
v	动词
vd	副动词
vn	名动词
vf	趋向动词

1.4 权重计算

本文提出的权重计算函数如下:

$$Weight(i) = W_{tf}(i) * W_p(i) * W_c(i) \quad (2)$$

其中, $Weight(i)$ 为候选词 i 的综合权重; $W_{tf}(i)$ 为 TFIDF 提取词 i 得到的权重; 计算方法是: $W_{tf}(i)=tf_i*\log(N/DF_i)$; $W_p(i)$ 为位置因子权重, 计算方法是: 根据提取词在文章中出现位置进行赋值. W_p=3, 出现在标题和正文中; W_p=2, 仅出现在标题中; W_p=1, 仅出现在正文中. $W_c(i)$ 为词性因子权重, 计算方法是: 根据提取词的词性来进行赋值. 提取词是名词性词汇 W_c=3, 动词性词汇 W_c=2, 其他词汇 W_c=1.

在本文算法中, 在计算词 $W_{tf}(i)$ 权重的基础上, 结合词性和词在文中出现的位置, 进行综合权重计算并由高到低排序, 从中提取 5 个词作为关键词, 仅提取 5 个词的原因是太多不利于检索, 太少不够全面. 在此过程中, 词性和词在文中出现的位置是权重计算的重要参数.

2 算法实现

本文算法 python 语言实现如下, 主程序中通过调用 keyword() 完成提取:

```
def keyword(title, text):
    keywords=analyse.extract_tags(text, withWeight=True)
    Wctext=""
    Wtf=[]
    for keyword in keywords:
    Keyword=keyword[0]
    Wctext+=Keyword
    #得到权重值
    wtf=keyword[1]
    #调用位置因子
    wp=localweight(Keyword, title, text)
    w=wtf*wp
    Wtf.append(w)
    #调用词性因子
    word=WordLabel(Wctext, Wtf)
  return word
```

以下为位置因子权重计算相关实现代码:

```
def localweight(word, title, text):
    wp=1
#在标题和正文中进行关键词查找, 找到权重乘以 3
if title.find(word)>0 and text.find(word)>0:
    wp*=3
else:
#仅在标题中, 权重乘以 2
    if title.find(word)>0 and text.find(word)<0:
        wp*=2
    else:
#关键词在文章中, 权重乘以 1
        wp*=1
return wp
```

以下为词性因子权重计算. 对特征词字符串 text 利用 psg.cut 完成词性获取. 相关实现代码如下:

```
def WordLabel(text, Wlist):
    words=psg.cut(text)
    List=[]
    RWlist=[]
    for i in range(0, len(Wlist)): #列表倒置
        RW=Wlist.pop()
        RWlist.append(RW)
    RL=RWlist
    for word in words:
        wc1=str(word.flag)
        wc=1
    #若为名词, 权重乘以 3
        if wc1.startswith('n'):
```

```
            wc*=3
        else:
            wc=1
            if wc1.startswith('v'):
    #若为动词, 权重乘以 2
                wc*=2
            else:
                wc*=1
    #其他词汇, 权重乘以 1
        if RL:
    #判断倒置列表是否为空
            W=RL.pop()
            Weight=wc*W
            tup=(wc.word, Weight)
            List.append(tup)
    L=sorted(List, key=by_Weight)
    #进行权重重新排序, 提取 5 个关键词
    s=''
    j=0
    for keyword in L:
        s+=keyword[0]+' '
        j+=1
        if j>4:
            break
    return s
```

3 实验分析

为完成算法的测试和评价, 测试数据采用的是预先提取的中国商务部日常新闻语料库, 分别采用传统 TFIDF 算法和本文所述 TFIDF 多特征改进算法完成关键词提取, 然后对结果进行评价.

3.1 评价标准

在结果评价中采用准确率 P, 召回率 R 以及 F 因子完成评价. 准确率是检索出的相关文档数与检索出的所有相关和不相关文档总数的比率, 也叫查准率; 检索出的相关文档数和文档库中所有检索到和未检索到的文档总数的比率, 也叫查全率. F 因子是两者的综合指标, 当 F 因子较高时, 则能说明试验方法比较有效. 计算方法如下所示:

(1) 准确率 Precision

准确率 P=提取出的正确信息条数/提取出的信息条数.

(2) 召回率 Recall

召回率 R=提取出的正确信息条数/样本中的信息条数.

(3) F 因子 F-measure:

F 因子=准确率 P*召回率 R*2/(准确率 P+召回率 R).

3.2 实验结果及分析

实验数据为预先提取的 4000 条新闻信息, 从中抽取 300 条新闻作为测试用例. 300 条新闻中包含了 100 条左右的短新闻, 并进行对应的分组. 将 300 条新闻分别分成 20 篇、50 篇、80 篇、100 篇和 150 篇, 目的是希望通过测试数据的增加, 验证本文所述改进算法是否会一直优于传统 TFIDF 算法.

利用百度搜索引擎进行关键词检索, 验证是否能检索到对应的新闻文章. 最后计算 TFIDF 算法和改进算法的准确率 P, 召回率 R, F 因子. 计算结果如表 2 所示.

表 2 两种算法评价指标对比

指标	算法及新闻篇数									
	TFIDF 算法					本文改进算法				
	20	50	80	100	150	20	50	80	100	150
准确率	0.3529	0.4318	0.473	0.433	0.5071	0.5789	0.8	0.7222	0.6146	0.7172
召回率	0.3	0.38	0.4375	0.42	0.4733	0.55	0.72	0.65	0.59	0.6933
F 因子	0.3243	0.4042	0.4546	0.4264	0.4896	0.564	0.7579	0.6842	0.602	0.705

为方便观察实验结果的变化, 对两种算法的准确率 (图 2)、召回率 (图 3) 和 F 因子 (图 4) 三项指标分别做图, 如下所示:

根据表 2 和图 2～图 4 可知: 在新闻篇数相同的情况下, 准确率 P, 召回率 R, F 因子三项指标结果均是 TFIDF 改进算法优于原本的单一算法. 证明本文所述 TFIDF 改进算法在一定程度上能够有效地提高关键词提取的准确度. 此外, 通过计算提取关键词的时间, 发

现改进的 TFIDF 算法与传统的 TFIDF 算法执行时间之差在 1 ms 以内.

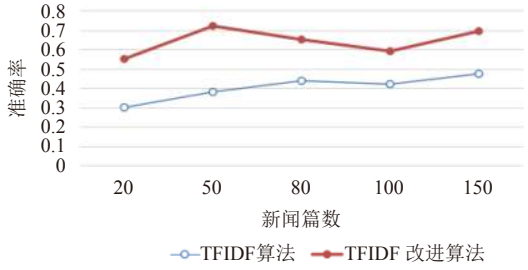

图 2 准确率折线图结果

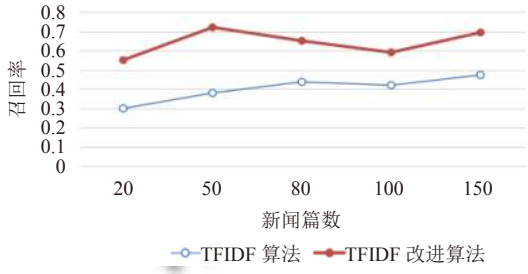

图 3 召回率折线图结果

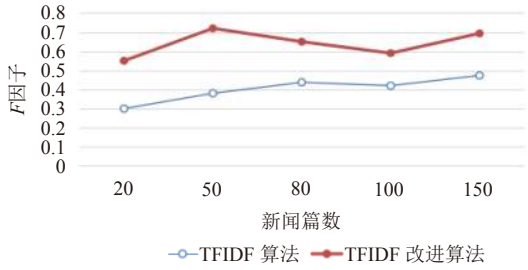

图 4 *F* 因子折线图结果

4 结论与展望

本文对传统的 TFIDF 算法进行了改进, 通过在关键词权重计算中引入词的位置因子和词性因子以降低 TFIDF 算法依赖词频而造成的影响. 同时, 利用 Python 语言完成了算法实现, 并对算法结果进行了实验验证.

实验结果表明, 本文所述算法能有效提升关键词提取的质量效果. 但由于测试条件有限, 本文评价标准的计算结果主要依赖于人工检索, 在测试语料的多样性和总体数量上也有一定局限性. 因此, 下一步将在条件许可时, 在大规模多样性测试语料集上进一步验证和改进算法.

参考文献

1 张瑾. 基于改进 TF-IDF 算法的情报关键词提取方法. 情报杂志, 2014, 33(4): 153–155.
2 谢晋. 基于词跨度的中文文本关键词自动提取方法. 现代物业·现代经济, 2012, 11(4): 108–111.
3 胡学钢. 基于词汇链的中文新闻网页关键词抽取方法. 模式识别与人工智能, 2010, 123(1): 45–51.
4 周炎涛, 唐剑波, 王家琴. 基于信息熵的改进 TFIDF 特征选择算法. 计算机工程与应用, 2007, 43(35): 156–158. [doi: 10.3321/j.issn:1002-8331.2007.35.047]
5 钱爱兵, 江岚. 基于改进 TF-IDF 的中文网页关键词抽取——以新闻网页为例. 情报理论与实践, 2008, 31(6): 945–950.
6 Yan Y, He L, Meng Q. Exploration and improvement in keyword extraction for news based on TFIDF. Energy Procedia,2011,(13):3551–3556.[doi:10.1016/S1876-6102(14)00454-8]
7 施聪莺, 徐朝军, 杨晓江. TFIDF 算法研究综述. 计算机应用, 2009, 29(6): 167–170, 180.
8 张建娥. 基于 TFIDF 和词语关联度的中文关键词提取方法. 情报科学, 2012, 30(10): 1542–1544, 1555.
9 袁津生, 毛新武. 基于组合特征的中文新闻网页关键词提取方法. 计算机工程与应用, 2014, 50(19): 222–226. [doi: 10.3778/j.issn.1002-8331.1212-0058]
10 黄轩, 李伟. 基于多特征的中文关键词抽取方法. 计算机与现代化, 2013, (4): 15–17.