基于加权 TextRank 的文本关键词提取方法

徐 立

（商丘职业技术学院软件学院 河南 商丘 476100） （中国科学技术大学苏州研究院 江苏 苏州 215000）

摘 要 为提升提取文本关键词的准确性，文中提出了一种文本关键词提取方法。该方法融合词频、词长、词语位置及词性等关键词提取影响因素，提出了候选关键词的权重公式；通过实验获取权重公式的相对最优权重系数；将权重公式应用到 TextRank 算法的候选关键词得分公式中，以提升提取文本关键词的准确性。通过实验对比了 OPW-TextRank 算法与 TextRank 算法对单文本关键词提取的准确率、召回率及 F 值，结果表明，OPW-TextRank 算法在窗口大小为 6 时，提取关键词的准确率高于 TextRank 算法。在以文本关键词提取为基础的自然语言处理系统中所提算法具有一定的实用性。

关键词 关键词提取，加权，词频，TextRank

中图法分类号 TP391.1 **文献标识码** A

Text Keyword Extraction Method Based on Weighted TextRank

XU Li

(School of Software, Shangqiu Polytechnic, Shangqiu, Henan 476100, China)

(Suzhou Research Institute, University of Science and Technology of China, Suzhou, Jiangsu 215000, China)

Abstract To improve the accuracy of keyword extraction, a text keyword extraction me-thod was proposed. This method combines the influence factors such as word frequency, word length, word position and word length, proposes the weight formula of candidate keywords. Then it obtains the relative optimal weight coefficient in the weight formula by experiment, applies the weight formula to the candidate keyword scoring formula of TextRank algorithm, and extracts the accuracy of text keywords. The accuracy, recall and F value of OPW-TextRank algorithm and TextRank algorithm in single text keyword extraction were compared through the experiment. The results show that the accuracy of OPW-TextRank algorithm is higher than that of TextRank algorithm when the window size is 6. It is useful in natural language processing keyword system based on text keyword extraction.

Keywords Keyword extraction, Weighting, Word frequency, TextRank

1 引言

文本的关键词可以反映出文本的主题思想，帮助读者快速对文本内容建立画像。提取文本的关键词可以被进一步应用于以下领域：文本的推荐、文本相似性度量[1]、文本的分类[2]、文本主题挖掘[3]、问答系统[4]、舆论热点的追踪[5]等。传统提取文本关键词的方法是组织领域专家进行人工标注，虽然准确率高，但效率低、成本高，完全无法适应当下海量文本的处理需求。

借助计算机技术是提高关键词提取效率、降低成本的有效途径。目前，用于提取文本关键词的方法可以粗略地分为有监督的和无监督的关键词提取。前者虽然在准确度上略高于后者，但需要一个足够量的人工标注关键词的文本集作为训练集，且容易产生过拟合现象，应用并不广泛。无监督关键词提取方法不需要训练集，大量研究通过改进提取方法来提升准确率，使之可与有监督的提取方法相媲美。

2 相关工作

常见的无监督的关键词提取方法如表 1 所列。

表 1 无监督关键词提取方法

方法	特点	典型算法
统计特征	词频、位置、长度等特征，易于理解、实现	TFIDF[6]
主题模型	文档-主题-词汇模型，相对复杂	LDA[7]
图模型	图论、矩阵等数学知识在关键词提取中的应用	TextRank[8]

单一使用上述任一种方法，在提取关键词的准确率上都不会有特别出色的表现，大量研究通过组合或改进上述方法来达到提升准确率的目的。李鹏等[9]通过将 Tag 值融入文本的图模型的节点和边的权重中，提出了一种 Tag-TextRank 算法，以提升提取文本关键词的准确率，但该方法只能提取有标签信息的网页的关键词，适用范围受到了很大限制。Ortega 等[10]利用标记过关键词的语料库进行训练，将 TextRank 算法变成了有监督算法，虽然提升了准确率，但带来了有监督方法的各种缺点；夏天[11]将词语的位置、频度等特性转化为权重，用于改进 TextRank 算法的词语得分公式，提升了算法的准确率。其与本文的工作最为相似，但考虑的影响词语重要性的特征过少且这些权重采用了简单的经验赋值。顾益军等[12]融合 LDA 和 TextRank，将单一文档信息和整体主题信息相结合进行关键词的提取，改善了关键词的提取效果，但代价是需要进行复杂的多文档主题分析。杨玥等[13]利用主题

徐 立(1982－)，男，硕士，讲师，主要研究方向为数据挖掘，E-mail：xuli1999@163.com。

模型分析和词频统计相结合的方法来提升提取关键词的准确率,代价同文献[12]。在提取关键词时,上述研究或通过方法的组合,或借助外部数据来提高准确率,在特定场景下,其效果可以与有监督的关键词提取方法媲美。如何在不依赖外部数据,通过改进算法来提高提取单文本关键词的准确率,是本文的研究重点。

本文提出的方法不依赖外部数据,将影响候选关键词重要性的因素(词频、词长、词语位置及词性)进行量化,搭配合理的系数获取候选关键词的最优综合权重,将此综合权重应用到 TextRank 算法计算关键词得分的公式中,提出了 OPW-TextRank 算法,以提升提取文本关键词的准确率。

3　OPW-TextRank 算法

3.1　评分公式改进

TextRank 算法的主要思想是:将文本转化为图模型,并通过式(1)迭代计算出词的得分,排名靠前的词可作为文本的关键词。

文本转化为图模型:将文本看成是句子集合 $T=\{S_1,S_2,\cdots,S_n\}$,任一句子 $S_i\in T$ 又可以看作词的集合 $S_i=\{W_1,W_2,\cdots,W_m\}$,构建图模型 $G=(V,E)$,其中 $V=S_1\cup S_2\cup\cdots\cup S_n$,当两个节点(词)共现于任一句子时,则节点间有边,否则无边。

$$Score(W_i)=(1-d)+d\times\sum_{j\in In(W_i)}\frac{1}{|Out(W_j)|}Score(W_j) \tag{1}$$

其中,$In(W_i)$是指向节点 i 的节点集合,$Out(W_j)$是节点 j 指向的节点集合。d 为阻尼系数,它原是 PageRank 算法随机游走概率,设置的初衷是为了防止那些没通过点击链接而跳转的页面吞噬用户向下浏览的机会,在文本图模型中,也存在没有任何指向的节点,通常情况下,d 值取 0.85。

在 TextRank 算法中,我们只考虑词与词之间的共现关系对重要性得分的影响,而事实上,除了词共现关系外,还有几个因素会影响词的重要性得分:词频、词长、词语位置及词性。将这 4 种影响因素量化,用于改进原有算法中的重要性得分公式,因此式(1)可以改进为:

$$Score'(W_i)=OPW(W_i)\times(1-d)+OPW(W_i)\times d\times\sum_{j\in In(W_i)}\frac{1}{|Out(W_j)|}Score'(W_j) \tag{2}$$

其中,$Score'(W_j)$为改进后节点 j 的得分;$OPW(W_i)$为节点 i 的最优化权重系数,是将词频、词长、词语位置及词性因素量化得到的,$OPW(W_i)$可以表示为:

$$OPW(W_i)=\alpha\times A(W_i)+\beta\times B(W_i)+\gamma\times C(W_i)+\delta\times D(W_i) \tag{3}$$

其中,$A(W_i)$,$B(W_i)$,$C(W_i)$,$D(W_i)$,α,β,γ,δ 分别为节点 i 的词频、词长、词语位置及词性的权重及权重系数,$\alpha+\beta+\gamma+\delta=1$。

3.2　影响因素的量化

1)词频

词频是衡量词在文本中出现次数的指标,利用词频提取文本关键词的经典算法是表 1 中提到的 TFIDF 算法。TF 是词在单文本中出现的频率。IDF 是反映词在整个文本集出现的频率(该值越大,表明词出现在整个文本集中的频率越低),算法认为一个词在单文本出现频率越大,在文本集出现的频率越小,则该词对单文本的影响越大,就越可能成为单文本关键词。由于 OPW-TextRank 设计的应用场景为单文本关键词的提取,尽可能地不依赖其它数据,因此本文只取词在单文本中的出现频率:

$$A(Wi)=\frac{Count(W_i,T)}{Count(T)} \tag{4}$$

2)词长

词语的长度不同,成为关键词的概率也不一样,可将关键词长度分为以下几类:1,2,3,4,5－7,7 以上,对下文实验环节标注的 1779 个关键词的长度进行的统计,结果分布如表 2 所列。

表 2　1779 个关键词的长度分布

关键词	1	2	3	4	5－7	7 以上
个数	9	631	437	455	148	99
占比/%	0.51	35.47	24.56	25.58	8.32	5.56

由结果可以看出,关键词的长度集中在 2,3,4,总占比超过 85%。在本文的改进算法中,暂将表 2 中的统计占比当成词长权重值 $B(W_i)$,在实际应用中,如有更大的关键词数据集,该权重值可以重新计算。

3)词语位置

词语首次出现在标题、摘要、首段、首句或其他位置,其成为关键词的概率也是不一样的,一般认为概率依次减小。根据词语出现在上述位置,可以给 $C(W_i)$分别赋值为:1,0.8,0.5,0.3,0.2。

4)词性

张建娥[14]对某文本集关键词的词性分布进行统计,如表 3 所列。

表 3　关键词的词性分布

词性	名词	动词	形容词	副词	其他
个数	8431	3405	1830	659	675
占比	0.562	0.227	0.122	0.044	0.045

由结果可以看出,关键词中,名词、动词、形容词的总占比超过0.9。借鉴张建娥[14]的研究成果,将表中的统计占比当成词性权重值 $D(W_i)$。

3.3　算法其他常用参数

在算法实际应用过程中,除了式(2)中的阻尼系数 d 外,还有以下参数会影响算法的效率和准确率:滑动窗口 win、迭代次数 k 和迭代阈值 t。

3.3.1　滑动窗口 win

在构建文本图模型时,需要计算两个词是否共现在一个句子中,来确定两个词对应节点间有无边相连。实际操作时,由于句子过长经过分词处理后,可能会产生数量较大的词集合,增大了词与词之间共现的机会,反映在图模型上为边过于稠密,会影响算法的效率及计算的准确率,因此引入滑动窗口的概念,将大的词集合切分,只有在窗口内共现的词才被认为有共现关系。

滑动窗口 win 可以表述为:句子 $S_i=\{W_1,W_2,\cdots,W_m\}$,对于特定 $win<m$,将句子划分为以下小集合:$\{W_1,W_2,\cdots,W_{win}\}$,$\{W_2,W_3,\cdots,W_{win+1}\}$,…,$\{W_{m-win+1},W_m,\cdots,W_m\}$,这些小集合就是滑动窗口。$win$ 值过大会造成图模型的边过于稠密,过小会造成边过于稀疏,都会影响算法运行的准确率,因此需要经过实验来确定最优值。

3.3.2 迭代次数 k 和迭代阈值 t

在式(2)中的阻尼系数取值小于 1 时,算法总能经过若干次迭代收敛,因此,应用算法时可以不设置迭代次数 k 和迭代阈值 t。但有时为了使算法达到某个可接受的结果后就终止迭代,可以通过设置迭代次数 k 和迭代阈值 t 来提前结束算法的运行。

3.4 提取单文本关键词的步骤

在使用 OPW-TextRank 提取文本关键词时要经过文本预处理、参数设置、转换图模型、初始化节点评分、迭代运算、取 TopN 等步骤,算法应用流程如图 1 所示。

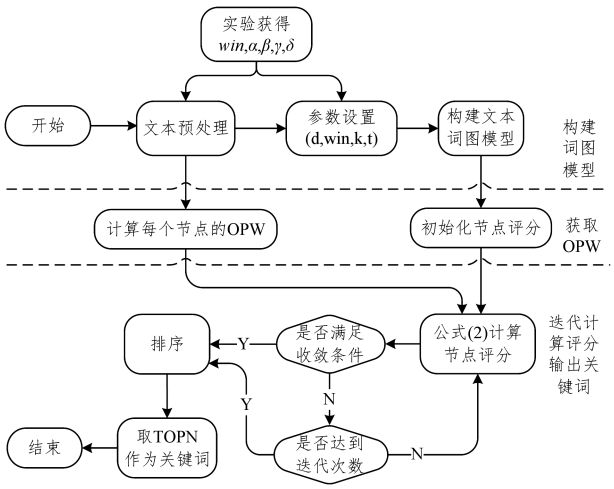

图 1 OPW-TextRank 算法的流程

文本预处理主要包括将文本按句子切词、去停用词。汉语与英文不同,词与词之间没有天然的分隔符号(空格),将句子变成词的集合必须经过中文切词。常用的切词工具有:IC-TCLAS、IKAnalyzer 和结巴切词。停用词(stop word)是指高频率出现在文本中且对文本语义信息没有贡献的一类词,比如“的”“了”“呀”“但是”这类语气词或连词、数字、英文字符等。去停用词的办法是建立停用词词库,遍历关键词候选词集合与停用词词库进行比对,比对成功就将其从候选词集合中删除。常用的停用词词库有百度停用词词库、哈工大停用词词库等。

参数设置是指上文中提到的阻尼系数 d、滑动窗口大小 win 等,滑动窗口大小的设置需要通过实验确定,以找到合适的 win 值。

运用式(2)迭代计算前,需要为每一个节点赋一个初始分值,文献[15]中证明了节点的初始分值不会影响最终的迭代结果,一般为节点初始赋值为 1。

相比原算法提取单文本关键词的步骤,改进后的算法增加了确定 OPW 系数的过程。当然,在文本预处理时,也要标注词频、词长、词语位置和词性等信息。

图 1 中,虚线部分是为了确定 $win,\alpha,\beta,\gamma,\delta$ 等参数的最优值,先凭经验为其设定初始值,然后通过实验得出最优值。实线部分是在测试集或工作数据集上运行算法的流程。

4 实验及分析

4.1 实验设计

实验环境:Intel 酷睿 i7 8700+16G 内存+win7 旗舰版。编程平台:Eclipse Java Oxygen+JDK1.8。

数据集:知网中 400 篇标注关键词的学术论文,共计 1779 个关键词,每篇论文平均 4.448 个关键词。其中,200 条用于实验获取最优权重系数、窗口值等,200 条用于测试两种算法的准确率。评价指标选用信息检索最常用的 3 个指标:准确率 P、召回率 R 及 F 值[16-18]。

$$P=\frac{ComputeWord\cap RealWord}{ComputeWord} \tag{5}$$

$$R=\frac{ComputeWord\cap RealWord}{RealWord} \tag{6}$$

$$F=\frac{2*P*R}{P+R} \tag{7}$$

$ComputeWord$ 是算法提取的关键词数量,$RealWord$ 是原文标注的关键词数量。实验采用与原算法对比的形式来分析实验结果。

4.2 实验实施

实验分为 3 个部分:1)运行 TextRank 和 OPW-TextRank 算法,确定滑动窗口值 win;2)运行 OPW-TextRank,确定式(2)的权重系数 $\alpha,\beta,\gamma,\delta$;3)在测试集上,设置上述参数和权重系数,运行 TextRank 和 OPW-TextRank 算法进行 P,R,F 值的比较。

4.2.1 确定滑动窗口值 win

在 OPW-TextRank 中,先假设词的 4 种得分影响因素的权重系数相等,即 $\alpha=\beta=\gamma=\delta=0.25$,滑动窗口值从 4 增加到 10,提取关键词个数为 6,获得原、新算法 F 值对比折线图,如图 2 所示。

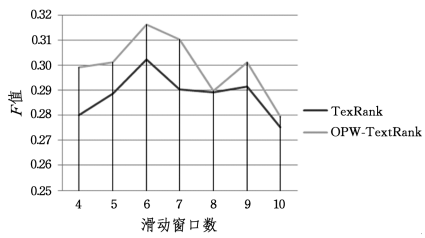

图 2 两种算法在不同滑动窗口下的 F 值折线图

由结果可以看出,两种算法均在滑动窗口 win 取 6 时获得最高的 F 值,后续在测试集上运行算法时,滑动窗口 win 取 6。

4.2.2 确定权重系数 $\alpha,\beta,\gamma,\delta$

查阅类似文献,其运行改进算法时,F 值都在 40 以下,因此在运行 OPW-TextRank 后,F 值越高越优,接近 40 即可认为获得了最优权重系数。滑动窗口 win 取 6,提取关键词个数从 4 增加到 10,对 $\alpha,\beta,\gamma,\delta$ 的以下 5 组取值进行实验:$E1$(0.25,0.25,0.25,0.25),$E2$(1,0,0,0),$E3$(0,1,0,0),$E4$(0,0,1,0),$E5$(0,0,0,1)。对应 F 值折线图如图 3 所示。

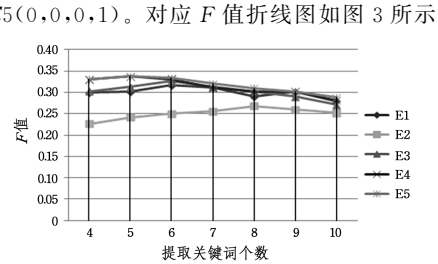

图 3 权重系数取不同值时的 F 值折线图

5 组取值中,$E1$ 相当于将几个影响因素视为同等重要,$E2$ 到 $E5$ 相当于分别只考虑词频、词长、词语位置和词性影响因素,以此来初步验证哪种影响因素权重的提高能更好地提高准确率。由结果可以看出:单考虑词频影响因素表现最

差,根据 TextRank 算法的固有特性,词频高的词更有机会与其他词共现,在图模型中表现为连接它的边多,更容易获取较高得分,相应的权重系数应当设置较低,来抵消这种双重重要性。单考虑词语位置和词性影响因素表现相对最优。后续调整取值,将词频因素权重降低,词语位置和词性因素权重提高,词长因素在 0.25 附近微调,经过反复实验获取权重系数为:$\alpha=0.091$,$\beta=0.221$,$\gamma=0.337$,$\delta=0.351$。此时式(3)转化为:

$$OPW(W_i)=0.091\times A(W_i)+0.22\times B(W_i)+0.337\times C(W_i)+0.351\times D(W_i) \quad (8)$$

4.2.3　测试集上运行原、新算法

在测试集上运行原、新算法,滑动窗口 win 取 6,OPW-TextRank 中最优权重系数按式(8)计算。在两种算法提取关键词个数从 4 增加到 10 时,对 P,R,F 值进行对比。

4.3　结果及分析

测试集上的运行结果如表 4 和图 4 所示。

表 4　两种算法的结果对比

提取关键词个数	算法	P	R	F
4	TextRank	0.3591	0.3022	0.3282
	OPW-TextRank	0.3666	0.3119	0.3370
5	TextRank	0.3408	0.3399	0.3403
	OPW-TextRank	0.3599	0.3383	0.3488
6	TextRank	0.3039	0.3463	0.3237
	OPW-TextRank	0.3226	0.3513	0.3363
7	TextRank	0.2787	0.3576	0.3133
	OPW-TextRank	0.2903	0.3643	0.3231
8	TextRank	0.2489	0.3813	0.3012
	OPW-TextRank	0.2639	0.3933	0.3159
9	TextRank	0.2285	0.4258	0.2974
	OPW-TextRank	0.2358	0.4315	0.3050
10	TextRank	0.1998	0.4308	0.2730
	OPW-TextRank	0.2101	0.4603	0.2885

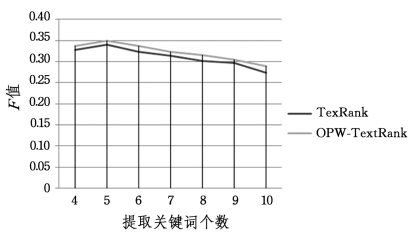

图 4　提取不同个数关键词的 F 值折线图

从表 4 的实验结果看出:两种算法随着提取关键词个数的增加,准确率均有所降低,说明得分最高的几个词是关键词的概率较大,在准确率上 OPW-TextRank 算法优于 TextRank 算法;随着提取关键词个数的增加,召回率逐步上升。

从图 4 两种算法的 F 值折线来看,取关键词个数为 4 和 5 时,两种算法的 F 值略有上升,大于 5 时整体逐步下滑,OPW-TextRank 的 F 值大于 TextRank 算法。很明显 OPW-TextRank 算法在单文本提取关键词时相对 TextRank 是有优势的,当然,前者在文本预处理时要标注更多信息,尤其需要实验获取权重系数,这些无疑增加了其复杂性,这是以后需要提升和改进的。

结束语　文本关键词的提取是一个应用广泛的领域,而 TextRank 算法又是处理文本关键词提取时应用最多的算法,很多研究都致力于改进 TextRank 算法,以提升关键词提取的准确率。本文提出的 OPW-TextRank 在提升关键词准确率上效果很明显,虽然增加了提取关键词的复杂性,但获得了较高的准确率,因此该算法有一定的应用价值。

本文在量化词语特征的权重时有些数据不准确,统计数据量不够大,比如统词长对词权重的影响时仅统计了 1779 个关键词,且这些关键词应用领域单一,如果能引入外部数据源,将统计数量增大,使领域变广,那么 OPW-TextRank 算法的准确率还有提升空间。

参 考 文 献

[1] 张璐,芦天亮,杜彦辉. 基于 WMF_LDA 主题模型的文本相似度计算[J/OL]. 计算机应用研究,2019(10):1-8.

[2] HASSAINE A,MECHETER S,JAOUA A. Text Categorization Using Hyper Rectangular Keyword Extraction: Application to News Articles Classification[C]// International Conference on Relational and Algebraic Methods in Computer Science. Springer International Publishing,2015:312-325.

[3] 曲靖野,陈震,胡铁楠. 共词分析与 LDA 模型分析在文本主题挖掘中的比较研究[J]. 情报科学,2018,36(2):18-23.

[4] ZHANG W N,MING Z Y,ZHANG Y,et al. Exploring Key Concept Paraphrasing Based on Pivot Language Translation for Question Retrieval[C]// Design Automation and Test in Europe. 2015:1-4.

[5] 夏火松,甄化春. 大数据环境下舆情分析与决策支持研究文献综述[J]. 情报杂志,2015,34(2):1-6,21.

[6] SALTON G,BUCKLEY C. Term-weighting approaches in automatic text retrieval[J]. Information Processing & Management,1987,24(5):513-523.

[7] BLEI D M,NG A Y,JORDAN M I. Latent dirichlet allocation [J]. Journal of Machine Learning Research,2003,3:993-1022.

[8] MIHALCEA R,TARAU P. TextRank:Bringing Order into Texts[J]. Emnlp,2004:404-411.

[9] 李鹏,王斌,石志伟,等. Tag-TextRank:一种基于 Tag 的网页关键词抽取方法[J]. 计算机研究与发展,2012,49(11):2344-2351.

[10] ORTEGA F J,VALLEJO C G. STR:A GRAPH-BASED TAGGING TECHNIQUE[J]. International Journal on Artificial Intelligence Tools,2011,20(5):955-967.

[11] 夏天. 词语位置加权 TextRank 的关键词抽取研究[J]. 现代图书情报技术,2013(9):30-34.

[12] 顾益军,夏天. 融合 LDA 与 TextRank 的关键词抽取研究[J]. 现代图书情报技术,2014(Z1):41-47.

[13] 杨玥,张德生. 中文文本的主题关键短语提取技术[J]. 计算机科学,2017,44(S2):432-436.

[14] 张建娥. 基于多特征融合的中文文本关键词提取方法[J]. 情报理论与实践,2013,36(10):105-108.

[15] SCHMIDHUBER J. Deep learning in neural networks:An overview[J]. Neural Networks,2015,61:85-117.

[16] CSOMAI A. Keywords in the mist:Automated keyword extraction for very large documents and back of the book indexing[J]. Unt Theses & Dissertations,2008.

[17] DOSTÁL M,JEZEK K. Automatic Keyphrase Extraction based on NLP and Statistical Methods[C]// Dateso 2011 International Workshop on Databases, Texts, Specifications and Objects. Pisek,Czech Republic,DBLP,2011:140-145.

[18] TIMONEN M,TOIVANEN T,TENG Y,et al. Informativeness-based Keyword Extraction from Short Documents[C]// KDIR. 2012:411-421.