第 38 卷 第 1 期
Vol.38 No.1

计 算 机 工 程
Computer Engineering

2012 年 1 月
January 2012

• 专栏 •

文章编号：1000—3428(2012)01—0001—04 文献标识码：A 中图分类号：TP391

基于语义的中文文本关键词提取算法

王立霞[1,2]，淮晓永[1]

(1. 中国科学院软件研究所基础软件国家工程研究中心，北京 100190；2. 中国科学院研究生院，北京 100049)

摘 要：为克服传统关键词提取算法局限于字面匹配、缺乏语义理解的缺点，提出一种基于语义的中文文本关键词提取(SKE)算法。将词语语义特征融入关键词提取过程中，构建词语语义相似度网络并利用居间度密度度量词语语义关键度。实验结果表明，与基于统计特征的关键词提取算法相比，SKE 算法提取的关键词能体现文档的主题，更符合人们的感知逻辑，且算法性能较优。

关键词：关键词提取；语义相似度；词语语义相似度网络；居间度；中文文本

Semantic-based Keyword Extraction Algorithm for Chinese Text

WANG Li-xia[1,2], HUAI Xiao-yong[1]

(1. National Engineering Research Center of Fundamental Software, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;
2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

【Abstract】In order to overcome the limitation of literal matching and lacking semantic concept of the traditional keyword extraction algorithm, this paper presents a Semantic-based Keyword Extraction(SKE) algorithm for Chinese text. It uses semantic feature in the keyword extraction process and constructs word semantic similarity network and uses betweenness centrality density. Experimental results show that compared with the statistic based keyword extraction algorithm, the keywords SKE algorithm extracted are more reasonable and can represent more information of the document's topic, and the SKE algorithm has a better performance.

【Key words】keyword extraction; semantic similarity; word semantic similarity network; betweenness centrality; Chinese text

DOI: 10.3969/j.issn.1000-3428.2012.01.001

1 概述

随着互联网及物联网的发展，更多的内容被数据化，数据的海量增长使得文本信息的分析与处理需求日益突显。目前，文本处理技术主要包括文档自动聚类、文档检索、文档自动摘要等。这些文本处理工作中的关键词提取是基础工作之一。例如，Web 文档聚类是对检索结果的聚类，将检索到的大量网页划分成一定的类别，然后提取类别的关键词作为类别的概括表示，最终将类别和关键词展示给用户，使用户能够快速定位期望的目标[1]。

针对海量的文档信息，人们提出了很多关键词自动提取方法，主要分为 3 类：(1)基于统计特征的方法，如词语频率统计；(2)基于词语网络的方法，根据一定规则将文档映射为词语网络，利用词语网络计算词语的关键度[2]；(3)基于语义的方法[3-4]，利用词语的语义特征提取关键词，如文献[3]引入同义词概念提高关键词提取的准确度。

在上述关键词提取算法中，基于统计特征的算法虽然操作简单，但是会忽略出现频率不高或在文档中位置不重要但对于文档具有关键意义的词语。基于词语网络的方法，目前主要是将高频词语以及它们在同一窗口(邻接、句子、段落等)的共现关系映射成词语网络[2]。词语网络的顶点集选取过程排除了低频词语，不能提取出对文档重要但频率不高的词语。根据词语的共现关系确定词语间的联系，缺乏语义理解，使得同主题但不在同一窗口的词语无法关联。基于语义的方法从语义角度判断词语的重要性，较符合人们的感知逻辑，也是目前关键词提取领域的一个研究热点。但目前在基于语义的关键词提取方法中，仅简单采用同义词和近义词匹配[3]，由于汉语文献作者使用语言的多样性，表达同一主题的关键词，大多不是同义词或近义词，使同主题的词语大部分未能得到语义关联，导致语义在关键词提取中不能发挥应有作用。

本文提出一种基于语义的关键词提取(Semantic-based Keyword Extraction, SKE)算法，以中文文本为处理对象，基于词语语义相似度将文档映射为一个词语语义相似度网络，结合社会网络理论，在词语语义相似度网络中使用居间度密度度量词语语义关键度，将词语语义关键度和词语的统计特征值加权获得关键词。

2 基本概念定义

2.1 词语语义相似度

词语间的语义相似度一般通过词语间的语义距离来描述[5]，一种常见的语义距离计算方法是根据同义词词典(thesaurus)[6]求 2 个词语编码的距离。本文采用现代汉语较常用的一部类义词典——哈工大的《同义词词林》扩展版，其中每个词有若干个编码，每个编码由 5 层代码和 1 位标志位描述，即编码 $Code_i$ 描述为 $Code_i=X_{i1}X_{i2}X_{i3}X_{i4}X_{i5}F_i$。5 层代码分别描述大类、中类、小类、词群和原子词群，1 位标志位为“=”、“#”或“@”，其中，“=”表示同义；“#”表示同类，属于相关词语；“@”表示词语自我封闭、独立，在词典中既没有同义词，也没有相关词。

基金项目：国家自然科学基金资助项目(90920010)；国家“863”计划基金资助项目(2008AA01Z145)

作者简介：王立霞(1986－)，女，硕士研究生，主研方向：中文信息处理，数据挖掘；淮晓永，高级工程师

收稿日期：2011-07-05 **E-mail：**lixia@nfs.iscas.ac.cn

定义 1 假设词语 W_1 在《同义词词林》扩展版中的编码有 m 个，分别为：$Code_{11}, Code_{12}, \cdots, Code_{1m}$，词语 W_2 的编码有 n 个，分别为：$Code_{21}, Code_{22}, \cdots, Code_{2n}$，则词语 W_1 和 W_2 的语义距离 $Dis(W_1, W_2)$定义为：

$$Dis(W_1,W_2)=\min_{i=1,2,\cdots,m;j=1,2,\cdots,n} Dis(Code_{1i},Code_{2j}) \tag{1}$$

根据文献[7]提到的概念层次树的深度问题：路径长度相同的 2 个结点，如果位于概念层次的越高层，其语义距离越大，即假设 2 个编码 $Code_1$ 和 $Code_2$ 从第 i 层开始代码不相同，其中，$1\leqslant i\leqslant 5$，则 i 所处的层次越高(i 值越小)，两编码间的语义距离越大。因此，本文中给不同的层次分配不同的语义距离权重，层次越高权重越大。定义权重数组 $weights=[w_1, w_2, w_3, w_4, w_5, w_F]$，其中，$w_1>w_2>w_3>w_4>w_5>w_F$，本文中 $weights$ 定义为[1.0, 0.5, 0.25, 0.125, 0.06, 0.03]。

定义 2 编码 $Code_1$ 和 $Code_2$ 的距离 $Dis(Code_1, Code_2)$定义为：

$$Dis(Code_1,Code_2)=\begin{cases}0 & Code_1=Code_2 \text{ 且 } F_1=F_2\neq\text{"\#"}\\ weights[5]\times init\_dis & Code_1=Code_2 \text{ 且 } F_1=F_2=\text{"\#"}\\ weights[i-1]\times init\_dis & Code_1 \text{和} Code_2 \text{从第} i \text{层开始编码不相同}\end{cases} \tag{2}$$

其中，$Code_1=X_{11}X_{12}X_{13}X_{14}X_{15}F_1$；$Code_2=X_{21}X_{22}X_{23}X_{24}X_{25}F_2$；$init\_dis$ 为自定义距离初始值。本文中 $init\_dis$ 取为 10。

定义 3 W_1 和 W_2 的语义相似度 $Sim(W_1, W_2)$定义为[5]：

$$Sim(W_1,W_2)=\frac{\alpha}{Dis(W_1,W_2)+\alpha} \tag{3}$$

其中，α 是一个可调节的参数，表示当相似度为 0.5 时的词语距离值。α 控制语义相似度 Sim 的取值范围，α 越大，语义相似度越不灵敏。本文中 α 取值为 5，语义相似度取值范围为 0.33~1。

2.2 词语语义相似度网络

目前，基于词语网络的关键词提取中，词语网络一般为词语共现网络[2]。词语共现网络构建过程为：限定一个窗口(邻接、句子、段落等)，若 2 个词语在同一窗口出现的频率大于设定阈值，则认为它们有联系存在“连接”。词语共现网络构建简单，但是不能确保同一窗口内的所有词语都有联系，而且未能体现词语间的语义关联。因此，本文根据词语间的语义相似度构建词语网络。

定义 4 设 D 为输入文档，预处理后的词语集合为 W，W_i 表示第 i 个词语，文档 D 对应的词语语义相似度网络图 G 定义为：

$$G=\{V, E\} \tag{4}$$

其中，V 表示图 G 的顶点集；V_i 表示 V 中第 i 个顶点；V 与 W 中元素一一对应，即 V_i 对应 W_i；E 表示图 G 的边集。

如果 2 个顶点的语义相似度大于一定的阈值，则在这 2 个顶点之间添加一条无向边，即：

$$E=\{(V_i, V_j) \mid V_i, V_j\in V, Sim(V_i,V_j)>\beta\}=\{(V_i, V_j) \mid V_i, V_j\in V, W_i, W_j\in W, Sim(W_i,W_j)>\beta\}$$

其中，$0<\beta<1$，β 越大，词语之间的语义关联要求越严格，则图 G 越稀疏，本文中 β 取 0.66。

2.3 居间度密度

社会网络描述社会实体(参与者)和他们之间的活动与关系。通过对社会网络的分析能够得到社会实体之间的各种关系和社会实体在网络中的作用以及网络的发展趋势。顶点中心性是社会网络分析的一个重要方面，顶点中心性有一系列计算方法，以顶点居间度为主。将图 G 视为社会网络，则图 G 中顶点居间度定义如下：

定义 5 顶点 V_i 的居间度 bc_i 定义为：

$$bc_i=\sum_{m,k=1}^{n}\frac{g_{mk}(V_i)}{g_{mk}} \tag{5}$$

其中，n 为图 G 的顶点数目；g_{mk} 表示顶点 V_m 和 V_k 之间的最短路径数；$g_{mk}(V_i)$表示顶点 V_m 和 V_k 之间的最短路径是否通过顶点 V_i，通过 V_i 则为 1，否则为 0。

定义 6 顶点 V_i 的居间度密度，指将图 G 中所有顶点的居间度集合平均划分成一定数目的区间后，顶点 V_i 的居间度所在区间的顶点密度。

在词语语义相似度网络中，通过大量实验发现文档中小部分主题无关词语的居间度非常大，若采用居间度作为词语语义贡献值，会将这部分词语误判为关键词。另外，在实验过程中发现：将居间度集合划分成一定数目的区间后，区间内词语越密集，即词语的居间度密度越大，该词语越可能是关键词。从文档 D 来看，文档围绕主题构建，与主题越相关，词语越密集，即与主题最相关的词语占的比例较大。并且主题相关词语与主题有语义相关性，所以他们彼此也有一定的语义相似度。词语语义相似度网络中，如果两词语语义相似，则在两词语之间添加一条边，这部分词语到其他顶点的最短路径情况也类似，即主题相关词语的居间度处于同一水平。所以，词语与主题越相关，词语的居间度密度越大。

根据上述分析，本文采用居间度密度作为词语的语义贡献值。居间度密度能够正确反映词语的语义关键程度，并能有效排除居间度很大但主题无关的词语。本文采用快速居间度计算方法[8]计算居间度。根据居间度计算居间度密度的算法步骤如下：

输入

V：图 G 的顶点集合，第 i 个顶点为 V_i。

bc：顶点集合 V 对应的居间度集合，V_i 的居间度为 bc_i。

s：bc 的区间划分个数，s>0，为可变参数。

c：区间个数增长的速度参数，c>0，为可变参数。当区间划分不能精确体现居间度集合 bc 的值分布情况时，需不断增加区间个数来细化 bc 的值分布情况，增加速度由 c 决定。

d：区间密度阈值，0<d<1，为可变参数。bc 的所有区间密度的最大值不小于 d 时，需增加区间个数重新划分居间度集合 bc。

max：对 bc 重新划分的最大次数，$max\geqslant 1$，为可变参数。当区间细化循环次数大于 max 时，程序立刻中止。

输出

Vd：顶点集合 V 的居间度密度集合，V_i 对应的居间度密度为 Vd_i。顶点对应词语，Vd 即为词语集合对应的居间度密度集合。

```
begin
(1)将 bc 平均划分成 s 个区间 Interval;
(2)计算 Interval 中各个区间的顶点数目占全部顶点数目的比例作为区间的密度，并保存到区间密度数组 IntervalDensity 中;
(3)maxratio := max(IntervalDenisty);
(4)如果 maxratio≥d:
   loop:= 1;     // loop 为 bc 的区间细化循环次数
   Refinement _BC; //寻找 bc 的最优区间划分细度;
(5)在 IntervalDensity 中查找每个顶点对应的居间度密度，并保存在顶点居间度密度集合 Vd 中;
(6) return Vd;
end
```

子算法 Refinement _BC 步骤如下：

begin
(1) s:= s×c;
(2)将 bc 平均划分成 s 个区间，并用 Interval 表示；
(3)计算 Interval 中各个区间的密度，保存到 IntervalDensity 中；
(4) maxratio := max(IntervalDenisty);
(5) loop:=loop+1;
(6)重复(1)~(5)，直到 maxratio<d 或者 loop>max;
end

不同的文档居间度分布情况不同，若采用相同的区间划分数目，会造成一些文档的居间度集合划分细度不够。因此，居间度密度算法根据文档动态调整 *bc* 的区间划分数目，动态调整策略如下：

(1)设置阈值 *d*，控制居间度集合的划分细度。算法运行过程中循环地判断当前居间度集合的划分细度是否满足区间密度阈值 *d*，如果不满足，则将区间个数增加到原来的 *c* 倍，对 *bc* 重新划分。

(2)设置中止参数 *max*,控制居间度集合的重新划分次数。在实验过程中发现：小部分文档的区间细化循环在可接受的时间和空间范围内无法停止，甚至造成程序无响应。通过对这些文档分析发现：这部分文档的居间度集合中存在小部分极大值或者极小值，在可接受的时间和空间范围内无法找到满足阈值 *d* 的最优区间划分。为此，算法设置了区间细化过程中止参数 *max*，当区间细化循环次数大于 *max* 时，程序立刻中止，保证程序的运行时间和空间都维持在可接受范围。

本文中各个参数取值为：*s*=10，*c*=5，*d*=0.8，*max*=6。

2.4　词语统计特征

定义 7　词语 W_i 在文档 *D* 中的词频 tf_i 定义为：

$$tf_i = \frac{n_i}{\sum_k n_k} \tag{6}$$

其中，n_i 是词语 W_i 在文档 *D* 中出现的次数；分母是文档中所有词语出现的次数总和。词频越大，词语越可能是关键词。

逆向文档频率(Inverse Document Frequency, IDF)是词语普遍重要性的度量。包含词语 W_i 的文档越少则 IDF 越大，表明词语 W_i 具有很好的类别区分能力。

定义 8　词语 W_i 在文档集合 *DS* 中的逆向文档频率 idf_i 定义为：

$$idf_i = \ln \frac{|DS|}{\left|\{d : d \ni W_i, DS \ni d\}\right|} \tag{7}$$

其中，|*DS*|为文档集合中的文档总数；分母是包含 W_i 的文档数目。

定义 9　词语 W_i 的词频-逆向文档频率(TF-IDF)$tfidf_i$ 定义为：

$$tfidf_i = tf_i \times idf_i \tag{8}$$

定义 10　词语 W_i 的词性值 pos_i 为 W_i 所属词性的重要度，定义为：

$$pos_i = \begin{cases} 0.5 & W_i\text{为形容词} \\ 0.3 & W_i\text{为副形词} \\ 0.6 & W_i\text{为名形词} \\ 0.6 & W_i\text{为成语} \\ 0.7 & W_i\text{为简称略语} \\ 0.6 & W_i\text{为习用语} \\ 0.3 & W_i\text{为动词} \\ 0.2 & W_i\text{为动语素} \\ 0.4 & W_i\text{为副动词} \\ 0.6 & W_i\text{为名动词} \\ 0.8 & W_i\text{为名词} \end{cases} \tag{9}$$

定义 11　词语 W_i 在文档中出现的位置 loc_{ij} 定义为：

$$loc_{ij} = \begin{cases} 0 & W_i\text{未在位置 } j\text{ 上出现过} \\ 1 & W_i\text{在位置 } j\text{ 上出现过} \end{cases} \tag{10}$$

其中，位置 *j* 取值为 1、2、3，分别表示标题、段首、段尾。标题、段首或段尾出现的词语比正文中的词语重要。

3　基于语义的中文文本关键词提取算法

本文在关键词提取中融入语义特征，提出由词语的语义贡献值和统计特征值共同确定的 SKE 算法。SKE 算法的逻辑结构如图 1 所示，主要由 4 个模块组成：文本预处理模块，词语语义贡献值计算模块，词语统计特征值计算模块，词语关键度计算模块。

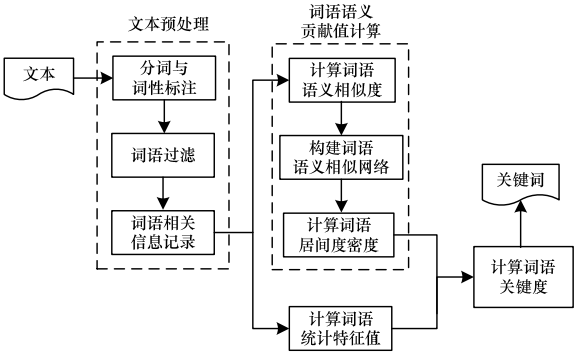

图 1　SKE 算法的逻辑结构

SKE 算法首先对输入的文本进行预处理。预处理后的结果将进入词语语义贡献值计算模块和词语统计特征值计算模块。词语语义贡献值计算模块负责计算词语居间度密度作为词语的语义贡献值。词语统计特征值计算模块负责计算词语的统计特征值。词语关键度计算模块将这 2 个模块的评分结果加权得到词语的关键度。最后根据词语的关键度输出最终结果。

SKE 算法处理步骤如下：

输入　文档 *D*

输出　文档 *D* 的关键词

(1)对文档 *D* 进行分词和词性标注，获得候选词语列表 *CandidateWords*。

(2)去除 *CandidateWords* 中的停用词后，保留形容词、副形词、名形词、成语、简称略语、习用语、动词、动语素、副动词、名动词和名词，获得词语集合 *W*。

(3)记录词语集合 *W* 中词语的词语文本、词语长度、词语位置以及词语词性。

(4)按 2.1 节的方法计算 *W* 中词语间的语义相似度。

(5)按 2.2 节中图 *G* 的定义，根据词语集合 *W* 和词语间的语义相似度构建词语网络 *G*。

(6)计算图 *G* 中所有顶点的居间度，得到居间度集合 *bc*。

(7)按 2.3 节中居间度密度算法，根据 *bc* 计算得到居间度密度集合 *Vd*。

(8)按 2.4 节的方法计算 *W* 中词语的统计特征值。

(9)将 *Vd* 和统计特征值加权获得词语的关键度，词语 W_i 的关键度计算函数为：

$$\begin{aligned} Score(W_i) = {} & Vdw \times Vd_i + \\ & Tw(\sum_{j=1}^{3} locw_j loc_{ij} + lenw \times len_i + \\ & posw \times pos_i + tfidfw \times tfidf_i) \end{aligned} \tag{11}$$

其中，Vd_i 表示 W_i 的语义贡献值；*Vdw* 表示语义贡献值权重；*Tw* 为统计特征值权重；loc_{ij} 表示 W_i 是否在位置 *j* 上出

现过；$locw_j$ 表示统计特征中位置 j 的权重，其中，j 取值为 1、2、3，代表的位置种类分别为标题、段首、段尾；len_i 表示 W_i 的词长；$lenw$ 表示统计特征中词长权重；pos_i 表示 W_i 的词性值；$posw$ 表示统计特征中词性权重；$tfidf_i$ 表示 W_i 的 TF-IDF 值，$tfidfw$ 表示统计特征中 TF-IDF 权重。

本文中各个权重取值为：Vdw=0.6，Tw=0.4，$locw_1$=0.5，$locw_2$=0.3，$locw_3$=0.3，$lenw$=0.01，$posw$=0.5，$tfidfw$=0.8。

(10)根据词语的关键度输出关键词。

假设：文档 D 中的词语个数为 n，预处理后的词语个数为 m，图 G 的边数为 e，文档集合中文档总数为 t，则 SKE 算法时间复杂度为：$O(n)+O(n)+O(m)+O(m^2)+O(m)+O(m\times e)+O(m)+O(t\times m)+O(m)+O(m^2)$，其中每一项分别对应 SKE 算法中的每一步。一般，$m<n$，$t<r\times n$，$e<k\times n$，其中 r 和 k 为整数常量，则整个算法的时间复杂度为 $O(n^2)$。SKE 算法的空间复杂度为：$O(1)+O(1)+O(1)+O(1)+O(m)+O(m+e)+O(1)+O(1)+O(1)+O(1)$，即 $O(n)$。

4 实验结果与分析

4.1 定性分析

为了检验 SKE 算法，本文结合 1998 年 1 月 1 日《人民日报》的文章《我爱逛农贸市场》分析 SKE 算法的关键词提取结果，并与基于统计特征的算法[9]提取的关键词进行对比。《我爱逛农贸市场》关键词提取结果为：SKE 算法提取的关键词为农贸市场、爱、鱼、乌骨鸡、羊肉、花生；基于统计特征的算法提取的关键词为农贸市场、逛、爱、变化、美、生活。

由上述结果可以看出，由于基于统计特征的算法没有进行文章主题分析，提取的关键词不能很好地体现文章主题。而 SKE 算法提取的关键词能够更加全面地覆盖文章的主题信息，并且 SKE 算法能够提取出低频在文档中位置不重要但是意义非常重要的关键词“乌骨鸡”，词频为 3，并且未出现在文章的重要位置，文章中有一部分是主要描写购买家禽中鸡肉的情况，“乌骨鸡”对于文章来说是个关键词；“花生”词频为 3，仅出现在文章的非重要位置，文章中有一部分是主要描写购买花生的情况，“花生”对于文章来说是个关键词。然而，使用基于统计特征的关键词提取算法仅能提取出高频并且处于文章中重要位置的词语。另外，SKE 算法还能够克服基于统计特征的关键词提取算法字面匹配的缺点，能够提取出同主题但是字面形式多变的词语，比如词语“鱼”。《我爱逛农贸市场》中对“鱼”采用了丰富多彩的字面形式进行描述，如“鱼儿”、“鲤鱼”、“带鱼”等 7 种词语，SKE 算法融入了语义的概念，因而能够将这些同主题的词语联系到一起，但是基于统计特征的关键词提取算法根据字面形式匹配，会将这些词语分开处理，使得关键词“鱼”不能被正确地提取出来。

由上述分析可以看出，本文的关键词提取算法能够提取出低频在文档中位置不重要但是意义非常重要的关键词，并且能够克服传统关键词提取算法字面匹配的缺点。本文算法提取出的关键词，能够全面地反映文章的主题信息，更加符合人们的感知逻辑。

4.2 定量分析

为了进一步测试 SKE 算法的有效性，笔者从人民日报 1998 年 1 月的语料库中选择 174 篇文档作为测试语料，将 SKE 算法和基于统计特征的关键词提取算法[9]进行对比实验。语料库共计 2 305 896 字，174 篇测试文档覆盖多种文体，包括新闻、杂文以及诗歌等。评价算法性能指标为查准率(P)、召回率(R)以及两者调和的平均值 F_1 测度值。

$$P=\frac{A}{A+B},\quad R=\frac{A}{A+C},\quad F_1=\frac{2PR}{P+R}$$

其中，A 为人工提取和自动提取均判断为关键词的数目；B 为人工提取为非关键词而自动提取为关键词的数目；C 为人工提取为关键词而自动提取为非关键词的数目。算法提取的关键词个数分别设置为 5、6、7。实验结果如表 1 所示。

表 1 SKE 算法和基于统计特征算法的性能对比

关键词提取个数	P		R		F_1	
	SKE 算法	基于统计特征的算法	SKE 算法	基于统计特征的算法	SKE 算法	基于统计特征的算法
5	0.765 2	0.632 2	0.535 6	0.442 5	0.630 1	0.520 6
6	0.851 4	0.701 1	0.596 0	0.490 8	0.701 2	0.577 4
7	0.889 2	0.698 9	0.622 4	0.489 1	0.732 3	0.575 5

可以看出，与基于统计特征的算法相比，SKE 算法在查准率、召回率和 F_1 测度值上均有明显提高，并且随着关键词提取个数增加，SKE 算法的优势越明显。实验数据证明了 SKE 算法的有效性。

5 结束语

本文提出了一种基于语义的中文文本关键词提取算法，利用词语语义构建词语语义相似度网络，在词语语义相似度网络中采用居间度密度度量词语的语义重要程度。本文算法提取的关键词能够体现文档语义层面的特征，而不仅仅只是统计特征值高的词语。实验结果证明了该算法的有效性，表明本文算法对于中文文本关键词提取的研究与应用具有一定的参考借鉴价值。下一步的工作是研究如何将特定领域的专家分类规则融入关键词提取中，以更好地利用已有的领域专家知识，进一步提高关键词提取的精度。

参考文献

[1] 姜亚莉, 关泽群. 用于 Web 文档聚类的基于相似度的软聚类算法[J]. 计算机工程, 2006, 32(2): 59-61.

[2] 张 敏, 耿焕同, 王照法. 一种利用 BC 方法的关键词自动提取算法研究[J]. 小型微型计算机系统, 2007, 28(1): 189-192.

[3] 程 涛, 施水才, 王 霞, 等. 基于同义词词林的中文文本主题词提取[J]. 广西师范大学学报: 自然科学版, 2007, 25(2): 145-148.

[4] 张 虹. 基于自动文本分类的关键词抽取算法[J]. 计算机工程, 2009, 35(12): 145-147.

[5] 刘 群, 李素建. 基于《知网》的词汇语义相似度计算[C]// 第三届中文词汇语义学研讨会论文集. 中国台北: [出版者不详], 2002.

[6] 张颖颖, 谢 强, 丁秋林. 基于同义词链的中文关键词提取算法[J]. 计算机工程, 2010, 36(19): 93-95.

[7] Agirre E, Rigau G. A Proposal for Word Sense Disambiguation Using Conceptual Distance[C]//Proc. of International Conf. on Recent Advances in Natural Language Processing. Tzigov Chark, Bulgaria: [s. n.], 1995.

[8] Brandes U. A Faster Algorithm for Betweenness Centrality[J]. Journal of Mathematical Sociology, 2001, 25(2): 163-177.

[9] 何新贵, 彭甫阳. 中文文本的关键词自动抽取和模糊分类[J]. 中文信息学报, 1999, 13(1): 9-15.

编辑 顾姣健