基于 CURE 算法的网页分块及正文块提取研究

王 超，徐杰锋
（中国石油大学(华东) 计算机与通信工程学院 计算机科学与技术系，山东 青岛 266000）

摘 要：研究基于 CURE 聚类的 Web 页面分块方法及正文块的提取规则。对页面 DOM 树增加节点属性，使其转换成为带有信息节点偏移量的扩展 DOM 树。利用 CURE 算法进行信息节点聚类，各个结果簇即代表页面的不同块。最后提取了正文块的三个主要特征，构造信息块权值公式，利用该公式识别正文块。

关键词：Web 信息抽取；聚类算法；页面分块；正文块提取
中图分类号：TP391 **文献识别码**：A **文章编号**：1674-7720(2012)12-0011-04

An approach based on CURE algorithm of Web page segmentation and information extraction

Wang Chao, Xu Jiefeng
(Computer Science and Technology Department, College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266000, China)

Abstract: This paper discusses an approach based on CURE algorithm of Web pages segmentation and text extraction rules. The main idea is to add attributes to nodes of a standardization DOM tree to convert it into the extended DOM tree with the information node offset. Subsequently, we use the CURE algorithm to cluster information nodes. And each result of the cluster represent different block of the page. Finally, we extracts three main features of the text block and construct information weights formula which can distinguish text blocks.

Key words: Web information extraction; clustering algorithm; page block; text block extraction

Web 信息抽取是以 Web 页面作为输入，最终得到正文信息的过程。它是人们利用互联网信息的有力保障。自 20 世纪 60 年代中期开始的信息抽取研究，至今成绩斐然，出现了一批具有代表性的方法和系统。

参考文献[1]构造的 WHISK 系统借助传统自然语言处理 NLP(Natural Language Processing)技术，由句子分析器将文本切割成多个实例，抽取规则以正则表达式的方式给出。该方法需要事先给出语料库，如果语料库选择不当，则会直接影响系统的正确性。参考文献[2]中采用了基于正则表达式的信息抽取方法。它针对不同的页面模板，构造 HTML 标记锚点模板库，并结合正则表达式，将标题、正文、关键字提取出来。在实际中，为了取得较好的效果，需要构造非常复杂的正则表达式，这也限制了该方法的应用。基于 HTML 结构分析的信息抽取也是经常被人采用的一个基本方法。该方法在信息抽取之前，首先将页面解析成一棵 DOM 树，通过自动或者半自动的方式产生抽取规则，并应用于 DOM 树上[3]。典型代表是 W4F(World Wide Web Wrapper Factory)系统[4]。它使用一个 HTML 解析器，将 Web 页面解析成树结构，通过自定义的一套 HEL 语言进行信息抽取，然后将结果保存到自定义数据结构 NSL(Nested String List)中。

就目前 Web 信息抽取发展趋势来看，无论哪种方法，都不能忽视页面固有结构。本文在结构研究的基础上，吸取前人方法的可取之处，提出用聚类算法进行页面分块、利用块特征识别正文块的方法。

1 层次聚类算法 CURE

层次聚类算法是聚类中常用的方法之一，它利用用户预期的聚类个数，进行数据集的层次分解，直至终结条件满足。主要有凝聚和分裂两个分支。

1.1 凝聚和分裂的层次聚类算法

凝聚的层次聚类是采用自底而上的策略，首先将数据集中的每个对象单独作为一个簇，然后依次合并簇，

直到满足结束条件。分裂的层次聚类与凝聚的层次聚类相反，采用自顶向下的策略。它首先将所有对象视为一个簇，然后再将它们逐渐细分为越来越小的簇，直至达到终止条件。

层次聚类算法虽然比较简单，但是算法的后一步总是在前一步新生成的簇上进行处理，且所进行的聚类是单向的，已经合并的点不能撤销，簇间也不能交换对象。如果在计算中间步骤做出错误的合并决定，最终可能会导致较差的结果。因此本文主要使用其改进算法CURE。

1.2 CURE 算法

CURE 算法摒弃了用单个质心表示一个类的传统方法，创新地用新生成的类的代表点计算该类与其余类间的距离，并设定一个收缩因子收缩或移动这些类。收缩因子的引入降低了噪音对聚类的影响，从而增强了CURE 处理孤立点时的健壮性。算法复杂度是 $O(n)$，因此也比较适合大型数据的聚类。图 1 为其算法流程。

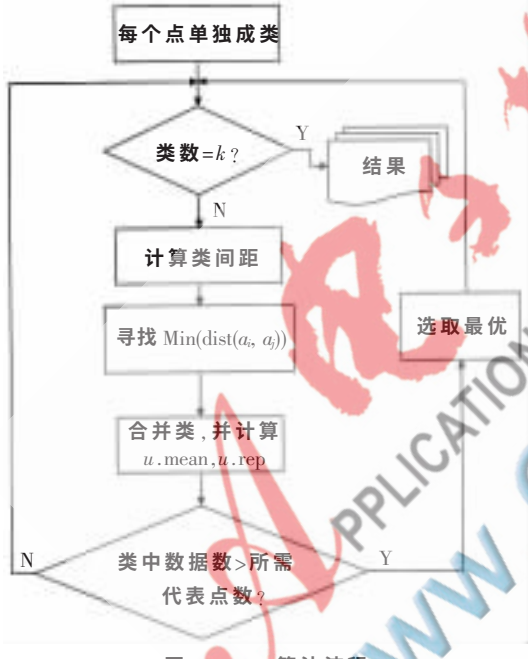

图 1 CURE 算法流程

下面介绍 CURE 算法中所用公式：两个聚类 u、v 之间的距离定义如式(1)所示：

$$\mathrm{dist}(u,v)=\min_{p\in u.\mathrm{rep},q\in p.\mathrm{rep}}\mathrm{dist}(p,q) \tag{1}$$

新产生的类的中心点按照式(2)计算，其中 w.mean 代表新生成的类的中心点，$|u|$、$|v|$ 分别代表该聚类 u、v 中所含数据项的个数：

$$w.\mathrm{mean}=\frac{|u|u.\mathrm{mean}+|v|v.\mathrm{mean}}{|u|+|v|} \tag{2}$$

新产生的类的代表点按照式(3)计算，其中收缩因子 ε 取 0.5，$w.\mathrm{rep}$ 代表新聚类 w 的代表点：

$$w.\mathrm{rep}=p+\varepsilon\cdot(w.\mathrm{mean}-p) \tag{3}$$

2 基于 CURE 算法的页面分块及正文块的提取

2.1 Web 页面规范化

由于 HTML 标准是松散的，以致很多 Web 页面的结构不严谨。因此页面清理首当其冲，主要包括规范化 HTML 代码(例如标签补全、嵌套错误修改)、清理 HTML 中的脚本代码、格式布局代码、空标签等，因为这些元素内部都不含有需要提取的正文信息。在处理之前将它们去除，可以提高系统效率。

2.2 DOM 树扩展

通过对 DOM 树结构的研究发现，语义上相同的块的父节点深度往往较大，也就是说，相同类别的信息节点一定有一个相同的父节点，且它在整棵 DOM 树中位于较深层。因此，可以对所有的信息节点进行遍历，在这个过程中，回溯寻找相邻信息节点的公共父节点。如果该父节点深度较小，极端情况下直至将要找到 DOM 的根时才发现它，则它们属于同一块的可能性较小。例如，对于图 2 中的示例，对信息节点 title 和 p1 进行比较时，直至 body 节点才发现它们的公共父节点，而 p1、p2、p3 在比较时，它们的公共父节点就出现在它们的上层，即 div 节点。那么它们是同一类信息节点的可能性就更大。

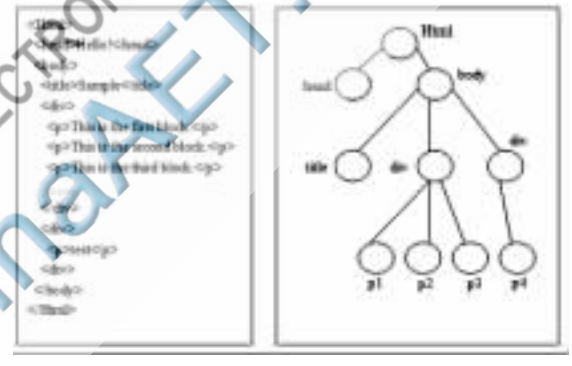

图 2 一个简单页面 HTML 源码及其 DOM 树

本文所提出方法的几个核心定义如下：

定义 1 信息节点：特指在一棵 DOM 树中，包含文本信息的叶子节点。同理，所有的布局节点、结构节点、修饰节点、脚本节点统称为非信息节点。

定义 2 非信息节点的权值：对于一棵深度为 n 的 DOM 树，对每一个非信息节点均有依附于该节点的一个权值，记为 W，则 W 的值如下，其中 l 为该节点所处的层：

$$W=\begin{cases}n & l=1\\ n-l+1 & l\neq 1\end{cases}$$

即：如果是根节点，其权值为 DOM 树的深度；如果是其他非信息节点，其权值为深度减去其所处的层次后加 1。

定义 3 信息节点的权值 T：在有序的 DOM 树中，依次从左子树开始，对信息节点进行序号标注，将这个序号作为该信息节点的权值，记为 T。

定义 4 偏移系数 α：度量信息节点相对其在 DOM 树中邻接的上一个信息节点的偏移程度，其值通过实验给出，它控制着两个信息节点能否归为一类的概率。α 的取值范围为：[0,1]。

定义 5 信息节点的偏移量 O_i：一个 DOM 树中的所有信息节点构成一个有序序列的集合 P，即：

$$P=\{b_1,b_2,\cdots,b_k\}$$

其中 b_i 代表信息节点，$i=1,2,\cdots,k$，k 为信息节点总个数

对于每一个信息节点 b_i 定义其偏移量 O_i 为：

$$O_i=\begin{cases}1 & i=1\\ O_{i-1}+1 & 1\leqslant i\leqslant k \text{ 且 } W_i\leqslant\alpha\cdot n\\ O_{i-1}+T\cdot W_i & 1\leqslant i\leqslant k \text{ 且 } \alpha\cdot n<W_i\leqslant n\end{cases}$$

定义 6 DOM 树中的所有信息节点均可以视为二维空间中的点：

$P_i=(O_i,C_i)$，其中 $i=1,2,\cdots,k$，k 为信息节点个数。

即：DOM 树种的所有信息节点均可以用偏移量以及字符数作为其属性。

通过以上逐条定义就得到了一棵扩展之后的 DOM 树，该树的非信息节点都带有权值，信息节点带有字符数和偏移量属性，这些属性是其属于不同信息块的特征的数量化表示。另外，通过定义 6 即可针对信息节点的属性进行 CURE 聚类。聚类过程中使用的距离采用欧氏距离。

2.3 正文块提取

在正确分块的基础上，通过提取块特征，进行正文块的识别工作。通过分析页面结构，可以得出正文块的以下几个显著特征：

(1)超链接较少：一个页面的正文块有可能会包含超链接，但是数量肯定较少。而其余的导航信息块、广告块等，包含的超链接数量肯定较多。

(2)含有较多长信息节点：正文块是信息的集中区域，故对应的信息节点也就较长。而对于其余噪声块，很少有多于正文块的信息量。

(3)含有较多数据记录条数：对于商品信息类页面，这些数据记录对应着一条条商品信息；对于新闻网页类页面，这些数据记录对应着各个文本段。

一个 Web 页面，非正文块也许也具备以上某条特征，但是同时具备 3 条特征的可能性却要小很多。例如，一个广告块存在很多条记录，它符合特征(3)，但是不符合特征(1)；一个评论块符合特征(1)，但是不符合特征(2)、特征(3)。另外，需要注意的是，3 个特征所表征正文块的能力是不同的，特征(2)、(3)对于表征一个正文信息块更加重要。

针对以上 3 个特征，给出信息节点超链比及信息块权值公式定义如下：

定义 1 信息节点的超链比：一个信息节点的超链比是该信息节点带有超链接的字符数(用 LinkTextLength 表示)与其所含全部字符数(用 TextLength 表示)的比值，即：

$$\text{Link}_i=\frac{\text{LinkTextLength}}{\text{TextLength}}$$

定义 2 信息块权值公式：一个信息块的权值用来表征该信息块作为正文块的可能性，它由该块内信息节点的超链比 Link_i、数据记录条数 N_1、长信息节点数 N_2 共同决定：

$$Rank=\partial\cdot N_1+\beta\cdot N_2+\gamma\sum_{i=1}^{N_2}(1-\text{Link}_i)$$

上式中，长信息节点的判断依据为该信息节点中所含字符数是否大于阈值 T。另外，∂、β、γ 分别是特征系数，代表各个特征的重要程度，$\partial+\beta+\gamma=1$ 且 $0<\partial,\beta,\gamma<1$。利用这个公式，可以分别计算各个块的权值，值越大，说明越符合正文块的特征，最后将具有最大权值的信息块判断为正文块。

3 实验测试

实验中依次选取新浪、腾讯、网易、人民网 4 个数据源的不同类型 Web 页面各 25 个，总共 100 个作为实验数据。系统使用正确率、召回率[5]进行系统性能判断，算法过程中使用到的参数都通过多次实验测定，具体如下：

①偏移系数 ∂ 取值为 0.7；

②CURE 算法中使用的预期聚类个数 K 依照不同源页面由人工指定。根据 CURE 算法提出者的取值建议，本实验中收缩因子 ε 为 0.5，代表点数 e 为 10；

③长信息节点阈值 T=50，即字符数大于 50 的信息节点判断为长信息节点；

④特征(1)的系数 ∂ 取 0.2，特征(2)、(3)系数等值，即 $\beta=\gamma=0.4$。

实验采用与 MDR 算法对比，结果如表 1 和图 3 所示。通过以上实验可以看出，本文提出的方法很好地进行了 Web 页面信息抽取工作，并且取得了较好的正确

表 1 本文方法与 MDR 算法的实验数据对比

	实际正文信息节点数	本文方法				MDR 算法			
		抽取正文信息节点数	正/误	查全率/%	查准率/%	抽取正文信息节点数	正/误	查全率/%	查准率/%
新浪	148	139	134/5	90.54	96.4	134	120/14	81.08	89.6
腾讯	118	121	113/8	95.76	93.39	131	109/22	92.34	83.2
网易	119	120	111/9	93.28	92.5	129	104/25	87.39	80.6
人民	91	92	85/7	93.4	92.39	112	83/29	91.21	74.1
总计	476	472	443/29	93.07	93.86	516	416/91	87.39	80.6

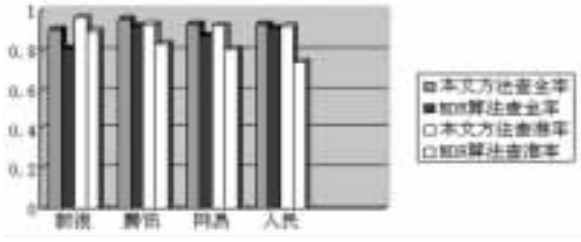

图 3 两种信息抽取方法比较

率和查全率。

互联网的快速发展使我们面临一个无所不有的巨大数据库，但是如何应用其中的数据却需要人们进一步研究。本文利用 CURE 算法进行页面分块，利用块特征进行正文块的识别。这种方法充分考虑到了页面结构所隐藏的规律，在实验中取得了较好的效果。

参考文献

[1] SODERLAND S. Learning information extraetion rules for semi-struetured and FreeText[J]. Maehine Learning,1999 (34):233-272.

[2] 刘华. 网页信息抽取及建库系统 C# 实习[J]. 计算机工程,2006,32(16):49-51.

[3] CRESCENZI V,MEECA G. RoadRunner: Towards automatic data extraction from large Websites[Z]. In Proeeedings of the 27th International Conference on Very LargeDatabase. Roma,Italy,2001:317-328.

[4] SAHUGUE A,ASAVAN F. Building intelligent Web applieations using light weight Wrappers[J]. Data Knowledge Engineering,2001,36(3):283-316.

[5] 李保利,陈玉忠,俞士汶.信息抽取研究综述[J].计算机工程与应用,2003(10):1-5,66.

(收稿日期:2012-03-20)

作者简介:

王超,男,1985 年生,硕士研究生,主要研究方向:数据挖掘。

徐杰锋,男,1964 年生,教授,博士,主要研究方向:Web 信息挖掘、数据库应用。