基于标题机器学习的网页分割方法

李进生[1] 乐惠骁[2] 童名文[2]
(武汉市广播电视大学现代教育技术中心 武汉 430033)[1] (华中师范大学教育信息技术学院 武汉 430079)[2]

摘 要 针对已有网页分割方法都基于文档对象模型实现且实现难度较高的问题,提出了一种采用字符串数据模型实现网页分割的新方法。该方法通过机器学习获取网页标题的特征,利用标题实现网页分割。首先,利用网页行块分布函数和网页标题标签学习得到网页标题特征;然后,基于标题将网页分割成内容块;最后,利用块深度对内容块进行合并,完成网页分割。理论分析与实验结果表明,该方法中的算法具有O(n)的时间复杂度和空间复杂度,该方法对于高校门户、博客日志和资源网站等类型的网页具有较好的分割效果,并且可以用于网页信息管理的多种应用中,具有良好的应用前景。

关键词 网页分割,标题,行块分布函数,块深度,机器学习
中图法分类号 TP37 **文献标识码** A

Novel Method of Web Page Segmentation Based on Title Machine Learning

LI Jin-sheng[1] LE Hui-xiao[2] TONG Ming-wen[2]
(Modern Education Technical Center, The Open University of Wuhan, Wuhan 430033, China)[1]
(School of Education Information Technology, Central China Normal University, Wuhan 430079, China)[2]

Abstract To solve the problem that it is difficult to implement the web page segmentation method based on document object model (DOM), a novel method was proposed through employing string model. The feature of the title of a web page is dug out by machine learning. Based on the found title, the web page is segmented. Firstly, the titles in web pages are picked up by the information of liner block function and title tag. Secondly, web pages are partitioned into content blocks by using the titles. Finally, the content blocks are merged by block depth information. It is proved that the complexity of algorithms in the method are O(n), and the method is suitable for web pages in the university portal, blog and resource web sites. The method is useful for many applications in web page information management, and it has a good prospect.

Keywords Webpage segmentation, Title, Liner block function, Block depth, Machine learning

1 引言

随着互联网应用的飞速发展,Web已成为全世界最大的开放、共享的信息库。网页是Web中信息组织与呈现的形式,一个网页中通常包含多个主题相对独立的内容块。网页分割是指将网页中独立主题的内容块划分出来的过程。网页分割技术广泛应用于搜索引擎优化、网页信息抽取和网页适配中,具有开阔的应用前景。

网页采用超文本标记语言(Hyper Text Markup Language,HTML)编写而成,该语言是一种半结构化的数据。本质上,网页是离散的文本内容和标签混杂在一起所组成的字符串。由于HTML本身缺乏语义表达能力,并且在实际中多数网页编写并没有完全遵循HTML的规范,这些问题给网页分割带来了很大的困难和挑战。目前,常见的网页分割方法大致分为三类:基于位置关系的分割方法、基于视觉信息的分割方法、基于样式的分割方法。其中被广泛采用的是Cai等研究者提出的VIPS算法[1],它是一种基于视觉信息的网页分割算法。但不管是上述的哪一种方法,它们都将网页构造成文档树,然后基于一定的准则判断树中节点的相似关系,最后将相似的节点组合为内容块,完成网页分割。这其中存在两个难点:1)如何定义节点的相似关系;2)如何减少文档树占用的内存空间。虽然,研究人员针对难点一做了大量的工作,且取得了一定的成果,但问题本身并没有得到很好的解决。因此,仍需要针对网页分割方法进行新的探索研究。

通过大量的观察发现,绝大多数网页的内容块都有标题。因此,标题可以作为网页内容块的天然分隔符。基于这一先验规律,提出了一种基于标题机器学习的网页分割方法。该方法将网页用字符串模型进行处理,通过利用行块分布函数和标题标签学习获取网页标题的特征,利用标题将网页分割为内容块,最后通过内容块组合得到最终的网页分割结果。该方法提出了一种新的实现网页分割的数据模型,是对网页分割新方法的有益尝试。

2 相关研究

国内外学者针对网页分割方法做了许多研究,并取得了丰富的研究成果。综合而言,目前提出的网页分割方法大致

本文受教育部人文社科基金资助项目:数字化学习资源无障碍适配决策模型研究(15YJA880062)资助。
李进生(1965-),男,教授,主要研究方向为终生学习理论、网络信息管理;**乐惠骁**(1995-),男,硕士,主要研究方向为网页信息处理;**童名文**(1975-),男,博士,教授,主要研究方向为数字化学习资源适配、教育资源管理,E-mail:tmw@mail.ccnu.edu.cn(通信作者)。

可以分为 4 类，具体分类及典型研究如下。

2.1 基于位置关系的分割方法

Chen 等[2]提出了一种利用网页页面的布局进行分块的方法。该方法将一个网页分成上、下、左、右和中间 5 个部分，再根据这 5 个部分的特征进行分类，在提取网页的内容后将其纳入到特征模版中，这样就把网页分成了几个特征块。这种方法适合于结构比较标准的网页，对于那些结构不是按照这种方式构造的网页则无法正确分割。另一种基于位置的网页分割方法建立在文档对象模型(Document Object Model，DOM)的基础之上。首先，找到网页中的特定标签(如〈table〉，〈p〉，〈img〉等)，再利用标签项将网页表示成一个 DOM 树结构；然后，根据各个模块在 DOM 树中的位置的不同将其进行分块。李文昊等提出了一种基于最优化理论的网页分割方法[3]。文献[4]和文献[5]也提出了基于 DOM 的网页分割算法。这类方法主要存在两个问题：1)DOM 树建树的过程和查找节点中的信息将花费大量时间和存储空间；2)由于部分网页没有完全遵循 HTML 规范，导致构建 DOM 树时出现错误。

2.2 基于视觉特征的分割方法

Cai 等提出了一种基于视觉特征的网页分割方法(VIPS)[1]，该方法首先将整个网页表示成一棵 DOM 树，根据颜色、大小等网页版面的特征，利用横竖线条将 DOM 树的节点所对应的分块从网页中分隔开，构成网页的标准分块。每个节点通过一致度(DoC)来衡量它与其他节点的语义相关性，从而将相关的分块聚集在一起。利用预先设定的一致度(PDoC)作为阈值控制分割粒度，当所有网页的 DoC 都不小于 PDoC 时，就可以停止网页分割。VIPS 综合考虑了 DOM 结构和视觉特征，是一种性能比较好的分割算法。但是 VIPS 算法也存在一些问题，如：1) 为了能够区分不同的分块，VIPS 算法根据实际的经验提出了很多启发式规则，对于网页元素缺乏统一的处理方式，规则逻辑不清晰；2)VIPS 算法基于 IE 浏览器实现，并没有对如何生成原始的页面分块进行描述，而这恰恰是基于视觉特征的网页分块算法的最关键的方面。因此，VIPS 算法无法适用于非 IE 浏览器。Zeleny 等提出了一种基于视觉信息和聚类的网页分割方法[6]，其最大的不同在于数据模型是呈现树而不是 DOM 树，从而降低了算法的时间复杂度，但该方法仍然基于树的数据模型。

2.3 基于样式的分割方法

孙晓辉等[7]提出了一种基于层叠样式表(CSS)的网页分割方法。该方法通过对网页进行解析和布局处理，提取出其中的 CSS 信息，并且使用重复模式检测和聚类的方法对生成的 CSS 树进行分割。此类算法也能较好地实现网页的分割，但是对于 CSS 信息较少的网页或者使用非 CSS 定义样式的页面则不能较好地进行分割。

3 基于标题学习的网页分割方案

已有的网页分割方法都将网页构造成 DOM 树结构，而建树过程既费时且占用存储空间。通过大量观察发现，大多数网页的内容块都有标题，标题是网页内容块的天然分隔符。基于这一先验规则，提出了基于标题学习的网页分割方法(以下简写为 BTLS)。

为表述清晰，对相关概念做如下定义：

定义 1(网页正文) 网页去除所有 HTML 标签后，留下的文字及空行组成的数据集。

定义 2(行块) 以网页正文中的行号为轴，取其下方 N 行，组成的数据块。其中 N 为行块厚度。

定义 3(行块长度) 行块中排除所有空白符(如：\n，\r，\t 等)剩余字符的总数。

定义 4(行块分布函数) 网页正文中行块标号到行块长度的映射[8]。

定义 5(块) 两个行块长度为 0 的行块之间最多非 0 行块组成的数据块。块中所有行块的长度都不为 0。

定义 6(行深度) 网页正文中行在原始网页 DOM 树中的深度。

定义 7(块深度) 块中第一个行的行深度。

定义 8(内容块) 网页中视觉整体或语义相关的正文块。

定义 9(标题块) 内容块中表示主题的块。

定义 10(正文块) 内容块中表示内容详情的块。

3.1 网页分割的流程

网页分割的基本思想是：利用网页中的行块分布函数和标题标签信息得到网页中的标题特征；再利用标题特征提取出网页中的标题，基于标题将网页分割为内容块；最后，以行深度为依据，组合内容块，完成网页分割过程。分割流程如图 1 所示。

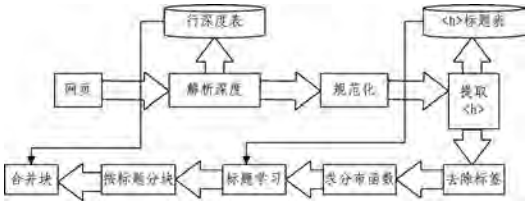

图 1 网页分割流程

网页分割流程的具体解释如下：

1)首先读取网页文档，解析其中每一行的行深度，组织成为一张行深度表。

2)对网页文档进行规范化。本方法中无需使整个文档符合 HTML 规范，只需要做两步处理：①若有一个 HTML 标签横跨多行，则将其规范化为一行内容；②规范化〈H〉标签对之间的内容，使其之间的信息的组织符合 HTML 规范。

3)提取〈H〉标签对之间的信息，组织成为一张〈H〉标题块表。

4)去除网页文档中的所有 HTML 标签，求网页内容行块分布函数。

5)结合行块分布函数和＜H＞标题块表，通过机器学习提取标题特征，并分割出文档中的所有标题块。

6)结合行深度表和标题块的信息，对标题块进行合并，并重新组织成为网页格式。

以下对网页分割流程中的关键步骤做详细说明。

3.2 行深度与块深度计算

虽然定义行深度时使用到了网页的 DOM 树，但在计算行深度时无需建立 DOM 树。行深度计算算法如图 2 所示。

1. 设定行深度值 depth 为 0，行号 i 为 0
2. 取网页文件的第 i 行，每遇到一个成对的 HTML 标签中的起始标签，depth 加 1；遇到一个成对的 HTML 标签中的结束标签，depth 减 1；遇到单个 HTML 标签中的左括号‘＜’，depth 加 1，遇到单个 HTML 标签中的右括号“/＞”，depth 减 1
3. 将 depth 保存在 rDepth[i]
4. 行号 i 加 1
5. 判断 i 是否到最后一行，如果程序结束，否则返回步骤 2

图 2 行深度计算算法

最后的结果存储在 rDepth[]数组中。以每个块的第一行的行号为索引查表 r 即可获得该块的块深度。

3.3　行块分布函数的计算

根据行块分布函数的定义，设计的行块分布函数算法如图 3 所示。

```
1. 设定行块厚度 sick
2. 计算网页正文中第 i 行块的长度，保存在 rbLength[i]
3. 行块标号 i 加 1
4. 判断 i 是否到倒数第三，如果是则结束程序，否则返回步骤 2
```

图 3　行块分布函数算法

最后的结果存储在 rbLength[]数组中。

3.4　标题块的提取

通过观察大量网页发现网页的行块分布呈现出一定的规律，即视觉上成为一块的内容块(CB)中的绝大多数由一个标题块(TB)和若干个正文块(PB)组合，如式(1)所示。

$$CB=TB+n*PB(n\geqslant 0) \tag{1}$$

网页行块分布呈现出上述规律的原因是网页内容是线性组织的。一般开发人员在编写网页时，总会将视觉整体或语义相关的内容代码组织在一起，便于代码的维护和可读性。因此，从行块的角度出发，多数网页可以表示成图 4 所示的模式。

```
(CB1)
TB
BB
TB
BB
...
(CB2)
TB
TB
BB
BB
...
(CB3)
TB
BB
BB
BB
...
```

图 4　网页内容块模式

换言之，网页内容块 CB 以标题块 TB 作为分隔符。因此，如果能够找出 TB，就可以实现网页分割。网页中提供了两种与 TB 相关的信息：1)TB 的行块长度；2)网页的标题标签〈H〉。综合应用这两种信息，可以很准确地提取出网页 TB。

观察网页的行块分布函数可以发现标题块的行块长度呈现出式(2)中的特征：

$$(0,0,n1,n1,n1,0,0,\cdots) \tag{2}$$

其中，$n1$ 是标题行长度。

呈现这种特征的块的一般编码模式如图 5 所示。

```
〈htmlTag1〉
〈htmlTag2〉
...
〈htmlTagN〉
Text
〈/htmlTagN〉
...
〈/htmlTag2〉
〈/htmlTag1〉
```

图 5　标题块的编码模式

对大量的网页分布函数进行统计分析发现，有上述行块分布特征的块很可能是 TB。为表达直观，举例说明，如图 6 所示。

图 6　网页标题实例

图 6 中的“文章分类”是标题，它的对应块中的编码如图 7 所示。

```
〈div id=“panel_Category” class=“panel”〉
〈ul class=“panel_head”〉
〈span〉
文章分类
〈/span〉
〈/ul〉
〈ul class=“panel_body”〉
```

图 7　标题块编码

取行块厚度为 3，该网页正文的行块分布函数为：0，4，4，4，0。

网页中还有一些非标题的短正文会给标题的特征检测带来干扰。在一个网页中，标题字符数通常比较相近，所以将字符数作为区分标题和短正文的特征值。因为不同的网页标题的字符数一般不同，所以需根据实际网页计算其标题字符数，而这一特征值能够通过遍历具体网页学习而获得。作为网页标题的字符数通常不会很长，满足式(3)：

$$C_1\leqslant n_1\leqslant C_2 \tag{3}$$

其中，C_1 和 C_2 为常数且 $C_1\geqslant 0$，$C_2\leqslant C$，$[C_1,C_2]$是一个网页中标题的可能字数区间；C 为一个经验值，是一个网页中标题的最大字符数，取值范围一般为[6，9]。因此，网页标题的行块长度是一个范围，但每个网页的范围不同。通过对两方面信息进行机器学习获得标题行块区间值：1)行块分布函数中满足($n1$，$n1$，$n1$，0)模式的块；2) HTML 语言的标题标签〈H〉。标签〈H〉是网页中着重强调文本标题的一类标签，用标签〈H1〉，〈H2〉，…，〈H6〉定义标题的 6 个不同的显示格式，本质上是为了呈现内容的结构性。

机器学习标题字符数的思路是：通过标题的行块特征在行块分布函数中找到满足标题特征的行块，以出现最频繁的块长度作为标题长度的一个参考值。通过标题标签找到所有标题行，计算所有标题长度的平均值作为另一个参考值。最后使用加权求和的方式计算出标题长度。通过设定一定容差值，确定网页标题字符数的上下边界值。具体算法如图 8 所示。在计算得到 C_1，C_2 之后，遍历集合 P，若标题块 $B\in P$ 且 B 中的 n_1 满足 $C_1\leqslant n_1\leqslant C_2$，则将 B 加入标题集合 H。

从实验中还发现，一些块也满足标题块特征，但是由于网页正文字数较少而没有被加入集合 P 中，这将出现标题块漏检问题。因此，在使用 C_1，C_2 确定标题块之后，再次遍历所有块，当发现某一个块满足：1)以该块为中心，在上下 3 个块

之内有标题块;2)该块对应的网页正文字数小于或等于 M(M 为一常数,取值在 10～15 之间),则该块也为一个标题块,并将其加入集合标题集合 H。至此,H 中保存了从网页中提取出的所有标题块。

1. 通过行块分布函数,找出网页正文中所有符合{0,$n1$,$n1$,$n1$,0,0,…}且 $n1 \leqslant C$ 的块,加入可能标题块集合 P
2. 找到网页中所有由〈H〉标签映射的且 $n_1 \leqslant C$ 的块,加入标题集合 H
3. 遍历集合 P 中的所有块,统计每个块的 n_1 并将其加入长度集合 L;遍历集合 L,找到其中重复出现的最多的行块长度 K_p
4. 遍历集合 H 中的所有块,统计每个块的 n_1 并将其加入长度集合 L_h;遍历集合 L_h,求 L_h 中所有元素的平均值 K_h
5. 若 H 非空,则标题块均值行块长度 $K=a_1 * K_p+a_2 * K_h$,c_1 和 c_2 是两个权重系数,$a_1<a_2$ 且 $a_1+a_2=1$。a_1 的取值范围一般为[0.3,0.4];若 H 为空,则 $K=K_p$
6. 设置一个容差值 δ,$C_1=\mathrm{Max}(K-\delta,0)$,$C_2=\mathrm{Min}(K+\delta,C)$,$\delta$ 的取值范围一般为[1,3]

图 8 网页标题字符数机器学习算法

3.5 内容块组合

如果单纯以标题块分割网页,则会产生分块过碎的问题,因此,需要将内容块进行组合,以达到更好的分割效果。在获取标题块的基础上,内容块组合分两步进行:

1)以标题块为分割边界,将标题块和正文块合并成为一个内容块。合并规则为:从块分布的起始开始向下遍历,若整个文档中的第一个块不是标题块,则从第一个块开始向下合并,直到遇到第一个标题块为止。该标题块之前的所有正文块合并为一个内容块。若块 $B(i)$ 为标题块,则顺序向下遍历,直到发现下一个块 $B(i+1)$,其满足块深度小于其上一个块(正文块或标题块),且是标题块;再将从 $B(i)$ 开始(包括 $B(i)$)到 $B(i+1)$ 的前一个块之间的所有块合并为一个内容块。然后从 $B(i+1)$ 开始,继续向下遍历,直到网页正文结束。

2)进行连续标题块的合并。合并规则为:重新遍历网页正文的块。当发现块 $BT(i)$ 是一个标题块时,顺序向下遍历,直到发现第一个内容块 $BC(i)$ 为止;再将以 $BT(i)$ 开始(包括 $BT(i)$)到 $BC(i)$ 的前一个块之间的所有块合并为一个内容块。然后从 $BC(i)$ 开始,继续向下遍历,直到网页正文结束。

4 数值分析

为证明该方法的网页分割效果和性能,从两个方面对 BTLS 法进行评价分析:1)在理论上分析该方法的实现算法的时间和空间复杂度;2)通过实验检验 BTLS 的有效性和性能。

4.1 理论分析

BTLS 处理网页的数据模型是字符串,其中的所有操作都通过线性遍历字符串即可实现。因此,理论上,BTLS 实现算法的时间复杂度为 $O(n)$,空间复杂度为 $O(n)$,具有较好的时空性能。而基于 DOM 的网页分割算法(如 VIPS)的时间复杂度和空间复杂虽然是 $O(n)$,但相对于 DOM 树的非线性结构,线性表的实现难度小很多。

4.2 实验分析

本次实验的硬件环境是:Pentium CPU 主频 3.06 GHz,2 GB 内存;软件环境为 Windows XP 系统;浏览器为 Google Chrome,编程语言为 C++。数据集取自高校门户、博客日志、咨询门户、资源网站和政府网站,从每类网站随机选取 10 个网页,共 50 个网页作为测试集。实验分为两步进行,首先进行网页分割实验。4 个典型网页的分割效果如图 9 所示。

(a)高校门户 (b)博客首页

(c)资源网站 (d)政府网站

图 9 分割效果

由图 9 可见,BTLS 能够较好地将网页分割为与视觉感受相符的内容块。为进一步评价分割效果,首先定义网页分割效果为 3 个等级,即:效果好、效果一般、效果差。然后选取 7 名志愿者,他们分别根据个人视觉感受对分割效果进行等级评定。最后统计所有网页的 3 个等级所占的百分比,以此衡量网页分割效果。实验统计结果见表 1。

表 1 网页分割效果统计

(单位:%)

网站类型	分割效果好	分割效果一般	分割效果较差
高校门户	80	10	10
博客日志	70	30	0
资讯门户	20	10	70
资源网站	70	20	10
政府网站	60	30	10

统计结果表明,BTLS 对高校门户、博客日志和资源网站网页的分割效果较好,70%的结果可以达到与主观视觉分割一致的效果;对政府网站的网页的分割效果一般,视觉分割一致率为 60%;对资讯门户网页的分割效果较差,视觉分割一致率只有 20%。

网页分割效果好的原因是:高校门户、博客日志、资源网站和政府网站的网页具有明显的标题,并且标题数量不多,便于机器学习得到网页的标题特征,标题可以被准确地提取出来。然而资讯门户类网页的分割效果较差,主要原因是:1)网页中标题过多,造成标题块混乱,机器难以学习标题特征;2)网页采用 Javascript 技术呈现,本方法由于针对 HTML 静态网页,因此无法处理包含 Javascript 的动态网页;3)网页中图片较多。本方法基于网页的标题信息,图片较多的网页含有的文字信息较少,能提取到的标题信息也较少,从而影响了分割效果。

然后进行性能评价实验。在该步骤中设计了两个实验,

一个用于统计 BTLS 实现网页分割的运行时间；另一个将其分割结果与 VIPS 进行比较。

在第一个实验中，将测试数据集中的每个网页进行 5 次分割，统计每种类型运行的最长时间、最短时间和中值时间。结果如表 2 所列。

表 2　运行时间

（单位：ms）

网站类型	最长时间	最短时间	中值时间
高校门户	32	18	22
博客日志	25	15	18
资讯门户	331	73	126
资源网站	86	18	26
政府网站	89	21	33

从实验数据可知，BTLS 对于高校门户、博客日志、资源网站和政府网站的网页均能在 40 ms 内完成分割，而对于资讯类网页的分割时间为 100 ms 左右。出现分割速度差异的原因主要是资讯类网页内容相对多且媒体形式丰富，数据量影响了程序的执行时间。

在第二个性能评价实验中，将 BTLS 与 VIPS 的分割结果进行比较，其中使用的 VIPS 也采用 C++语言实现[9]。对比性能指标为准确率、召回率和 F 值。首先，计算测试数据集中每个网页的 3 个指标值；然后，按类型计算 3 个指标的平均值；最后，对比两种方法在 3 个指标上的表现，以比较它们进行网页分割的性能。在此实验中，借鉴信息检索中的核心评价指标，对网页分割性能指标定义如下[10]：

准确率 P＝正确分割块数/分割总块数　　(4)

召回率 R＝正确分割块数/实际网页总块数　　(5)

调和平均值 $F=2*P*R/(P+R)$　　(6)

在实验中，VIPS 的分割阈值 $PDoC$ 设为 5 或 6，以避免分割过粗或过细。实验结果见表 3。

表 3　网页分割性能比较

	VIPS			BTLS		
	P	R	F	P	R	F
高校门户	0.62	0.53	0.57	0.83	0.86	0.84
博客日志	0.77	0.65	0.70	0.76	0.65	0.70
资讯门户	0.41	0.22	0.29	0.53	0.41	0.46
资源网站	0.63	0.75	0.68	0.78	0.67	0.72
政府网站	0.66	0.52	0.58	0.62	0.77	0.69

实验结果表明，对于高校门户、博客日志、资源网站和政府网站的网页，BTLS 分割网页的 F 值较高，达到 0.69 以上；其中高校门户类网页的分割性能最好，F 值达到 0.84。这一结果与文中所做的主观评价一致。另外，BTLS 实现的网页分割在 F 值指标上均优于 VIPS，尤其是在高校门户类网页上的优势更加明显。其原因在于高校门户类网页标题明确，且标题字数相近，BTLS 方法能够准确学习得到网页的标题字数特征，从而准确提取出网页标题，实现高准确度分割。但高校门户网页的视觉特征并不明显，多数页面不同的块的底色相同，这影响了 VIPS 的分割效果。对于所有的网页，BTLS 程序运行顺利，没有出现异常终止现象。但 VIPS 程序处理包含大量 javasript 的网页时经常出现错误提示，尤其在处理资讯门户类网页时，处理时间较长，且经常由于异常而终止程序。两种方法的鲁棒性不同的原因可能是因为其数据模型不同。BTLS 的数据模型是字符串，是一种线性模型。VIPS 的数据模型是树，属于非线性模型。两种模型的实现难度和空间复杂度的差异对程序的鲁棒性有较大影响。最后，从实验中还发现，BTLS 能够实现带折叠属性页网页的分割，而 VIPS 只能分割展示出的属性页内容，对于隐藏的属性页内容不做处理。这也与两种方法的数据模型相关，BTLS 的字符串模型将所有内容线性排列，依次处理，所以对属性页的所有内容都做处理；VIPS 的 DOM 树模型跳过了属性页的兄弟节点，所以没有处理隐藏内容。

结束语　基于网页标题机器学习的思想，提出了一种新的网页分割方法。该方法能够对多数的静态网页实现快速、有效的逻辑块分割。本方法的优点在于使用字符串数据模型，在 $O(n)$ 的时间复杂度和空间复杂度下实现高效的网页分割。该方法可作为搜索引擎、网页文档分类与聚类、信息抽取、信息重构与主题信息采集的预处理操作，具有广泛的应用前景。

尽管该方法具有一些优点，但实验结果表明，它对于标题较多、标题字数超出经验值范围以及采用 Javascript 技术实现内容布局的网页的分割效果一般甚至较差。因此，在接下来的研究中，将针对这 3 种情况进行方法改进与优化，使其具有更强的通用性。

参 考 文 献

[1] CAI D, YU S, WEN J R, et al. VIPS: a vision-based page segmentation algorithm: MSR-TR-2003-79[R]. Microsoft Technical Report, 2003.

[2] CHEN Y, XIE X, MA W Y, et al. Adapting web pages for small-screen devices[J]. IEEE Internet Computing, 2005, 9(1): 50-56.

[3] 李文昊，彭红超. 基于视觉特征的网页最优分割算法[J]. 计算机科学，2015，42(11)：284-287.

[4] 王琦，唐世渭，杨冬青，等. 基于 DOM 的网页主题信息自动提取[J]. 计算机研究与发展，2004，41(10)：1786-1791.

[5] HATTOR G, HOASH I K, MATSUMOTO I K, et al. Robust-Web Page Segmentation for Mobile Terminal Using Content-Distances and Page LayoutInformation[C]// Proceedings of the Sixteenth International World Wide Web Conference (WWW 2007). 2007.

[6] ZELENY J, BURGET R, ZENDULKA J. Box clustering segmentation: A new method for vision-based web page preprocessing[J]. Information Processing and Management, 2017, 53(3): 735-750.

[7] 孙晓辉，刘建，王劲林，等. 基于 CSS 的网页分割算法[J]. 网络新媒体技术，2008，29(9)：46-51.

[8] 陈鑫. 基于行块分布函数的通用网页正文抽取[EB/OL]. http://www.cnblogs.com/loveyakamoz/archive/2011/08/17/2143446.html.

[9] CAI D. ViPS: a Vision based Page Segmentation Algorithm [EB/OL]. http://www.cad.zju.edu.cn/home/dengcai/VIPS/VIPS.html.

[10] ARULJOTHI S, SIVARANJANI S, SIVAKUMARI S. Web Page Segmentation for Small Screen Devices Using Tag Path Clustering Approach[J]. International Journal on Computer Science and Engineering, 2013, 5(7): 617-624.