计算机科学
Computer Science
ISSN 1002-137X,CN 50-1075/TP

《计算机科学》网络首发论文

题目： 基于可视块的多记录型复杂网页信息提取算法
作者： 王卫红，梁朝凯，闵勇
网络首发日期： 2019-08-12
引用格式： 王卫红，梁朝凯，闵勇．基于可视块的多记录型复杂网页信息提取算法．计算机科学. http://kns.cnki.net/kcms/detail/50.1075.TP.20190812.1321.043.html

网络首发时间：2019-08-12 15:16:36
网络首发地址：http://kns.cnki.net/kcms/detail/50.1075.TP.20190812.1321.043.html

基于可视块的多记录型复杂网页信息提取算法

王卫红 梁朝凯 闵 勇

(浙江工业大学计算机科学与技术学院 杭州 310023)

摘 要 网页具有丰富的内容和复杂多变的结构，现有的网页信息提取技术解决了单记录型简单页面的信息提取问题，但是对于多记录型复杂页面的信息提取效果往往不佳。文中提出了一种全新的基于可视块的复杂网页信息自动化提取算法（Visual Block Based Information Extraction，VBIE），通过启发式规则构建可视块与可视块树，然后通过区域聚焦、噪声过滤及可视块筛选，实现了对复杂网页中数据记录的提取。该方法摒弃了以往算法对网页结构的特定假设，无需对 HTML 文档进行任何人工标记，保留了网页的原始结构，且能够在单页面上实现无监督的信息提取。实验结果表明，VBIE 的网页信息提取精确度最高可达 100%，在主流搜索引擎的结果页面和社区论坛的帖子页面上的 F1 均值分别为 98.5%和 96.1%。相比目前方法中在复杂网页上提取效果较好的 CMDR 方法，VBIE 的 F1 值提高了近 16.3%，证明了该方法能够有效解决复杂网页的信息提取问题。

关键词 Web 数据抽取，Web 挖掘，数据记录提取，网页数据提取，结构化信息

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/jsjkx.190200346

Multi-recording Complex Web Page Information Extraction Algorithm Based on Visual Block

WANG Wei-hong LIANG Chao-kai MIN Yong

(College of Computer Science and Technology，Zhejiang University of Technology，Hangzhou 310023，China)

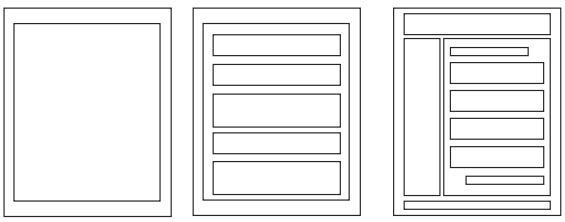
Abstract The webpage has rich content and complicated and varied structure. The existing webpage information extraction technology solves the information extraction of the single-recording simple page, but the information extraction effect of the multi-recording complex page is often poor. This paper proposed a new visual block based information extraction algorithm, named visual block based information extraction (VBIE).By constructing visual blocks and visual block trees, and through heuristic rules, regional focus, noise filtering and visual block filtering, data record extraction is realized for complex web pages. Compared with other existing methods, this method abandons the specific assumptions of the previous algorithm on the structure of the webpage, does not need to manually mark the HTML document, preserves the original structure of the webpage, and can realize unsupervised information extraction on a single page. The experimental results show that VBIE's web page information extraction accuracy is up to 100%, and the average value of F1 on the results page of the mainstream search engine and the post page of the community forum are 98.5% and 96.1%. Compared with the current method CMDR, the F1 value of VBIE is improved by nearly 16.3%, which proves that the method can effectively solve the information extraction task of complex web pages.

Keyword Web data extraction, Web mining, Data record extraction, Web page data extraction, Structured information

1 引言

互联网的出现为人们提供了快捷、高效的信息获取方式。根据 2019 年 2 月的第 43 次《中国互联网络发展状况统计报告》[1]，截至 2018 年底，我国的网站数量为 523 万，网页数量为 2816 亿，并且这些数据还在持续性地高速增长中，海量的网页使得互联网成为了最大的公开数据源。

为了适应人们的阅读习惯，网页中的各个组件和模块通常会按照一定的视觉规律排列，即网页布局。根据复杂程度，可以将其分为几种类型，如图 1 所示。单记录型页面如图 1(a)所示：一个页面中仅包含一个数据记录。常见的单记录型页面包含新闻页面、文章页面等，其正文信息占据了页面的显著位置，网页的噪声信息较少。以往的方法聚焦于此类型的页面，并获得了较好的结果。多记录型页面如图 1(b)所示：数据记录在网页中以列表形式依次排列，页面中的数据记录列表占据了最主要的部分。常见的多记录型页面包括目录页面、搜索结果页面等。目前人们接触最多的为多记录型复杂页面，如图 1(c)所示：页面中存在多种语义块，它们分布在数据记录列表所在的区域周围。该类型的页面包括社区论坛页面以及单页面应用。



(a)单记录型页面 (b)多记录型页面 (c)多记录型复杂页面

图 1 网页的 3 种常见布局

本文受浙江省自然科学基金（LY17G030030, LGF18D010001, LGF18D010002）资助。

王卫红（1969），男，硕士，教授，主要研究方向为空间信息服务、网络技术与安全；**梁朝凯**（1994），男，硕士，主要研究方向为网页信息提取；**闵 勇**（1981），男，博士，副教授，主要研究方向为社交网络分析、网络数据挖掘，E-mail：myong@zjut.edu.cn（通信作者）。

Fig.1 Three common layouts for web pages

本文主要面向多记录型页面和多记录型复杂页面，在充分利用网页内容信息、DOM 结构信息以及视觉信息的基础上，提出了一种更为健壮、高效的单页面网页信息提取方法 VBIE。

2 相关工作

目前主流的网页信息提取方法可分为 3 类，即基于网页文档的提取方法、基于 DOM 结构信息的提取方法、基于视觉信息的提取方法。

基于网页文档的提取方法将 HTML 文档视为文本进行处理，适用于处理含有大量文本信息且结构简单易于处理的单记录网页，或者具有实时要求的在线分析网页应用。这种方式主要利用自然语言处理相关技术实现，通过理解文本语义、分析上下文、设定提取规则等，实现对大段网页文档的快速处理。其中，较为知名的方法有 TSIMMIS[2]，Web-OQL[3]，Serrano[4]，FAR-SW[5]和 FOREST[6]，但这些方法由于通常需要人工的参与，且存在耗时长、效率低的弊端，不适用于多记录型页面。

基于 DOM 结构信息的方法将 HTML 文档解析为相应的 DOM 树，然后根据 DOM 树的语法结构创建提取规则，相对于以前的方法而言有了更高的性能和准确率。 W4F[7]和 XWRAP[8]将 HTML 文档解析成 DOM 树，然后通过组件化引导用户通过人工选择或者标记生成目标包装器代码。Omini[9]，IEPAD[10]和 ITE[11]提取 DOM 树上的关键路径，获取其中存在的重复模式。MDR[12]和 DEPTA[13]挖掘了页面中的数据区域，得到数据记录的模式。CECWS[14]通过聚类算法从数据库中提取出自同一网站的一组页面，并进行 DOM 树结构的对比，删除其中的静态部分，保留动态内容作为信息提取的结果。虽然此类方法相对于上一类方法具有较高的提取精度，且克服了对大段连续文本的依赖，但由于网页的 DOM 树通常较深，含有大量 DOM 节点，因此基于 DOM 结构信息的方法具有较高的时间和空间消耗。

目前比较先进的是基于视觉信息的网页信息提取方法：通过浏览器接口或者内核对目标网页预渲染，然后基于网页的视觉规律提取网页数据记录。Cai 等率先提出了经典的 VIPS[15]算法：首先从 DOM 树中提取出所有合适的页面区域，然后根据这些页面和分割条重新构建 Web 页面的语义结构。作为对 VIPS 的拓展，ViNT[16]，ViPER[17]，ViDE[18]也成功利用了网页的视觉特征来实现数据提取。CMDR[19]为通过神经网络学习多记录型页面中的特征，结合基于 DOM 结构信息的 MDR 方法，挖掘社区论坛页面的数据区域。与上述方法不同，VIBS[20]将图像领域的 CNN 卷积神经网络运用于网页的截图，同时通过类 VIPS 算法生成视觉块，最后结合两个阶段的结果识别网页的正文区域。Gogar 等[21]则是加入了上下文关系来划分网页的视觉区域。

本文所提出的 VBIE 方法，在基于网页视觉信息的基础上进行改进，摒弃了对网页结构的特定假设，能挖掘更多维度的网页信息，实现无监督的网页信息提取。VBIE 基于网页的复合信息构建可视块和可视块树，获得网页的正文区域之后，通过可视块筛选得到区域内的数据记录块。该方法无需多个同域同源的网页，且对多记录型页面具有较好的提取效果。

3 网页信息提取模型

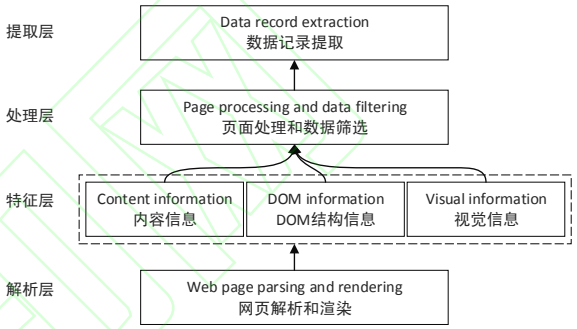


图 2 四层数据提取模型

Fig.2 Four-layer data extraction model

通过对相关工作的研究，可以将网页信息提取总结为如图 2 所示的四层模型，自底向上分别是解析层、特征层、处理层以及提取层。

解析层：获取网页文档及资源，对网页进行资源解析和渲染，提供脚本运行环境等。VBIE 属于基于视觉信息的方法，在这一层将网页资源交由浏览器或者浏览器内核进行解析和渲染，获得页面的视觉呈现。

特征层：通过解析层提供的数据访问接口获取网页的复合信息，如内容信息、DOM 结构信息以及视觉信息。VBIE 将网页的这 3 类信息组合为网页的复合信息，并以此构建网页的可视块和可视块树。

处理层：基于上一层得到的特征信息，对网页数据进行处理。VBIE 采用了基于视觉规律的区域聚焦来获得网页的正文区域，并根据区域内可视块之间的相似性进行聚类，标记其中的数据记录和噪声信息。

提取层：提取上一层被标记为数据记录的部分，并获得这些数据记录的数据项。

如图 3(a)所示，VBIE 在解析层获得页面并对其渲染，通过启发式规则构建网页的可视块和可视块树。图 3(b)展示了网页在区域聚焦后的状态，正文区域被挖掘出来作为处理层的作用范围。图 3(c)则是网页经过局部噪声过滤后的状态图，经过聚类获得数据记录簇，如图 3(d)所示。最

后，在提取层完成网页数据记录和数据项的提取。

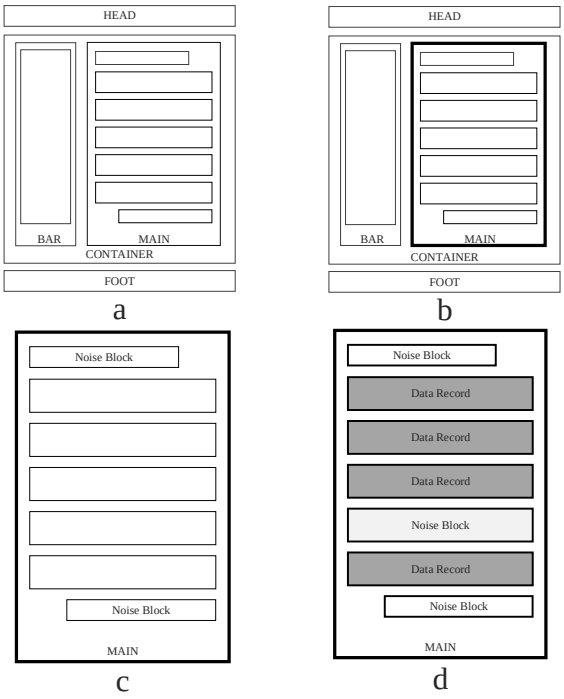


图 3 网页数据提取状态图

Fig.3 Web page data extraction state diagram

4 特征层：可视块与可视块树

4.1 网页的复合属性

网页包含 3 种不同属性的信息，分别是内容信息、DOM 结构信息以及视觉信息。本文将这 3 类信息进行组合，称为网页的复合信息，如表 1 所列。

内容信息作为网页的主体，包含了网页文档中的标签、文本、媒体、链接等数据，该部分信息可以由 HTML 文档通过自然语言处理直接获得。

DOM 结构信息是网页的骨架，由 HTML 文档转换得到。DOM 结构信息包含了 DOM 节点的深度、路径、上下文等信息。

视觉信息则是网页的呈现，需要经过 Web 浏览器内核对网页资源进行解析和渲染。视觉信息包含了文字、盒模型、布局、背景等信息。有效地利用网页的复合信息，可以帮助网页信息提取方法更精准地定位网页中的数据区域，提取网页中的数据记录，并挖掘其数据属性。

表 1 可视块的复合属性表

Table 1 Composite property table for visual blocks

属性	分类	指标	说明
内容信息	标签	HTMLTag	如 <p><a> <table>等 HTML 语义标签，通常成对出现
	文本	ContentText	DOM 节点内嵌的文本
		ContentLength	文本长度
	链接	LinkNum	链接数量，即有效锚标记的数量
	媒体	ImageNum	图片数量
		MultimediaNum	多媒体数量，包含音频和视频
DOM 信息	深度	DOMDepth	节点在 DOM 树中的层级
	路径	DOMPath	节点与根节点之间所形成的路径
	上下文	DOMChildrenNum	节点的直接子节点的数量
		DOMSiblingsNum	节点的相邻节点的数量
视觉信息	文字	Font(Color \ Style \ Size \ Weight \ Family)	字体颜色、字体类型、字体大小、字体粗细、字体系列
	盒模型	Box(Width \ Height \ Padding \ Border \ Margin \ Sizing)	元素宽度、元素高度、元素内边距、元素边框、元素外边距、元素的大小计算模式
	布局	Layout(Display \ Position \ RelativeLeft \ RelativeTop \ AbsoluteLeft \ AbsoluteTop \ Overflow \ Verticalalign)	元素的呈现样式、元素的定位、元素是否浮动、元素外边距边界与其父元素的边界之间的偏移（元素的相对偏移）、元素外边距边界与页面的边界之间的偏移（元素的绝对偏移）、元素的内容溢出规则、元素内的内容对齐方式（水平和垂直）
	背景	Background(Color \ Image \ Position \ Repeat \ Size)	背景颜色、背景图像、背景图像的绘制起始位置、背景图像的重复规则、背景图像的绘制尺寸

4.2 可视块构造算法

经过对 HTML 文档的解析和转化，可得网页的 DOM 树。由于目前网页的数据量过于庞大且结构复杂，HTML 文档中包含大量嵌套的标签对，其 DOM 树同样具有复杂的结构和繁多的节点。基于这样庞大的 DOM 树来进行网页信息提取，往往会导致算法耗时过长，且效率低下。针对以上问题，本文参考了 VIPS 算法所提及的视觉块，不直接将 DOM 树运用于网页信息提取，而是根据由 DOM 树结合网页的视觉呈现生成形式上更为简洁的可视块（VisualBlock，VB）。

为了将 DOM 转换为网页的可视块，本文提出了基于启发式规则的可视块构造算法。首先，依照网页的视觉特征制定了 5 条针对可视块提取的启发式规则，如表 2 所列；然后，遍历网页 DOM 树，并将其中符合启发式规则的 DOM 节点提取为可视块。VIPS 算法自顶向下进行网页的处理，需要耗时较长的分隔条检测，以及可能会破坏网页语义结

构的内容结构重构；本文所提出的可视块构造算法自底向上沿着 DOM 树构造可视块，且更为简洁高效，不影响网页的内容或是布局。值得注意的是，可视块依赖网页的复合信息构建，同时可视块本身也具有如表 1 所列的各类复合属性。

表 2 可视块的提取规则

Table 2 Visual block extraction rule

规则	规则说明
R_1	如果当前节点的宽高不满足至少大于当前页面的基准文字大小，则该节点不是可视块节点
R_2	如果当前节点有且仅有一个子节点，则两个节点合并为一个单元来判别是否为可视块节点
R_3	如果当前节点的文本内容为空或者仅包含空白字符，则该节点不能作为可视块节点
R_4	如果当前节点在网页中不可视，即透明、被完全遮盖或者其他情况，则该节点不能作为可视块节点
R_5	如果当前节点在布局上脱离了文档流，则该节点不能作为可视块节点

4.3 构建可视块树

为了减少算法的空间复杂度和时间消耗，同时保留可视块之间的上下文关系，本文基于可视块在 DOM 树上的分布，将可视块重新组织为树状结构，即可视块树（Visual Block Tree，VBT）。可视块树的构建步骤如下：

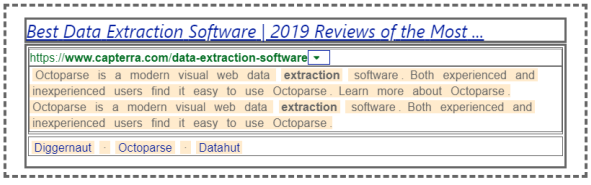
1）由 DOM 树的根节点开始，将根节点对应的可视块作为可视块树的根节点；

2）自顶向下对 DOM 树执行广度优先遍历，其具体步骤为由一个已访问的 DOM 节点出发，以层为顺序访问所有该 DOM 节点下面的子节点；

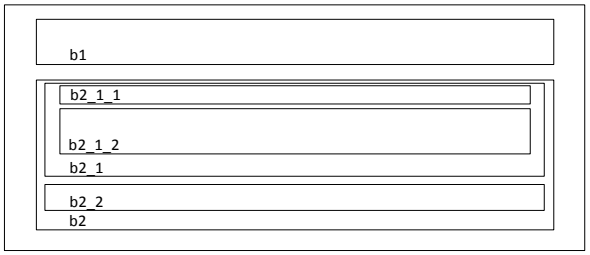
3）判断每一个 DOM 节点是否能够提取可视块，并依据可视块之间的层级关系，将所有可视块组织为可视块树。

与网页的 DOM 树不同，可视块树 VBT 有以下几个特征：1）VBT 中的总节点数更少；2）VBT 中具有视觉信息的每一个节点，都对应着网页中的一个矩形视觉区域；3）VBT 上存在父子关系的节点对，对应的网页矩形区域具有嵌套或者包含的关系。

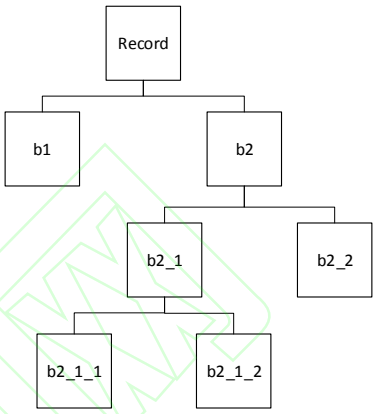
图 4 举例说明了从网页中构建可视块树的过程。



(a)网页语义块



(b)可视块分布模型图



(c)可视块树结构图

图 4 可视块树的构建示意图

Fig.4 Construction of the visual block tree

图 4(a)展示了一个网页元素的样例，该网页元素包含标题、文本、链接等子元素及 DOM 节点。图 4(b)展示了由该网页元素生成的可视块分布模型，并非所有的 DOM 节点都对应于一个可视块，如空节点、分隔符以及不可见 DOM 节点都不能作为可视块，而短文本、标题、链接等细粒度的 DOM 节点则被合并到其父亲节点或祖先节点所对应的可视块中。图 4(c)则展示了由前者生成的可视块树，其相比 DOM 树更为简洁明了，有效地降低了后续操作的复杂度。经观察，可视块树中的可视块具有如下规律:

1）如果可视块 *A* 是可视块 *B* 的父节点，那么块 *A* 对应的区域将包含块 *B* 对应的区域；

2）如果可视块 *A* 与可视块 *B* 在可视块树上具有相同的深度，那么块 *A* 与块 *B* 对应的区域相互隔离，不会发生重叠。

以上规律可用于下文提及的区域聚焦过程，帮助本方法更好地获取网页的正文区域。如果可视块 *A* 或者 *B* 的布局属性脱离了文档流，以上的规律将会失效，但是这些规律仍旧适用于绝大多数不同类型的网页。

5 处理层：页面处理和数据筛选

在经过特征层的处理之后，可以得到网页的复合信息，利用复合信息构建网页的可视块和可视块树，接下来需要在提取网页信息之前对网页进行处理。值得注意的是，本方法并没有直接从深度网页中标记数据记录，而是先进

行了区域聚焦，找到所有数据记录的最小数据区域，然后通过局部噪声过滤及可视块筛选，标记该区域内的数据记录。

5.1 区域聚焦

为了捕获用户的注意力，并让用户能高效地浏览页面，网页中的数据记录往往被聚集在一个矩形区域内，这种区域被称为数据区域，而挖掘数据区域的过程则称为区域聚焦。由于不同网页之间的 DOM 树结构差异会导致网页的编辑距离阈值不稳定，本方法并不依靠 DOM 树结构挖掘网页的数据区域，而是依靠视觉信息来识别数据区域。

与其他研究不同，本文基于数据区域的概念提出了区域聚焦，用于探索网页中包含所有数据记录的最小边界，即网页的正文区域。网页的正文区域对应于可视块树上包含网页所有数据记录的最小子树，所以提取正文区域能够有效地缩小网页信息提取的范围，减少噪声信息的干扰。正文区域基于数据区域而来，具有与数据区域一致的视觉特征。本文按照以下步骤进行网页正文的区域聚焦。

首先，遍历可视块树上的所有可视块，选择其中同时满足以下条件的可视块作为网页正文候选区域可视块。

1）可视块在网页布局上靠近网页中心。正文区域通常在网页中占据显著位置，如图 5 所示，其正文区域在离网页中心略偏右侧的位置。本文选择计算可视块的中心偏移量比率来判断可视块是否靠近网页的中心。以网页左上角为原点构建坐标系，X 轴为水平方向，Y 轴为垂直方向，每一个可视块的边界平行于轴。$OffsetLeft$和$OffsetTop$分别是可视块的左侧边和上侧边距离页面边界的距离，$BlockWidth$和$BlockHeight$代表了可视块的宽度和长度。

$$BlockX = OffsetLeft + BlockWidth/2 \tag{1}$$

$$BlockY = OffsetTop + BlockHeight/2 \tag{2}$$

其中，$BlockX$为可视块的垂直中线与页面边界的距离，$BlockY$为可视块的水平中线与页面边界的距离。由此可得可视块的中心坐标为$(BlockX，BlockY)$。，同理可得页面的中心坐标为$(PageX，PageY)$，则可视块的中心偏移量$Offset$以及可视块的中心偏移量比率B_o的计算分别如式（3）与式（4）所示。本文为中心偏移量设定了阈值T_o，小于这个阈值的可视块位于网页的显著位置。

$$Offset = \sqrt{(PageX - BlockX)^2 + (PageY - BlockY)^2} \tag{3}$$

$$BlockY = OffsetTop + BlockHeight/2 \tag{4}$$

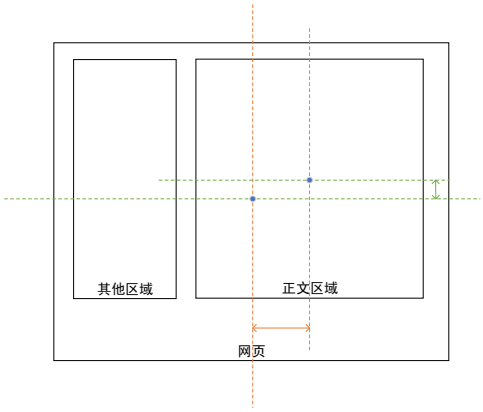


图 5 可视块的中心偏移

Fig.5 Center offset of the visual block

2）可视块在页面上表现为具有一定大小的面积。相对于整个网页而言，正文区域通常具有较大的视觉面积。本文通过计算可视块的面积大小比率，判断可视块是否符合作为正文区域的视觉大小。如式（5）所示，可视块的面积$Area_{block}$通过可视块宽度$BlockWidth$与可视块高度$BlockHeight$相乘得到，同理可得页面的面积$Area_{page}$。通过式（6）可得到可视块与页面的视觉面积比率B_a，本文为其设定了阈值T_a，如果可视块的视觉面积比率大于该阈值，则可视块被标记为拥有足够大的视觉区域。

$$Area = Width * Height \tag{5}$$

$$B_a = Area_{block}/Area_{page} \tag{6}$$

3）可视块具有足够丰富的内容。网页的正文区域包含了网页最主要的信息，通常具有较长的篇幅。内容是否丰富，可由文本的长度进行量化。本文将可视块的内容文本长度与页面的内容文本长度进行比较，从而判断可视块的内容是否足够丰富。如式（7）所示，$ContentLength_{block}$和$ContentLength_{page}$分别是可视块的文本长度和页面的文本长度，B_c为可视块的内容长度比率。本文为其设置了阈值T_c，如果可视块的内容长度比率B_c大于该阈值，则该可视块拥有足够丰富的内容。

$$Area = Width * Height \tag{7}$$

基于以上 3 个条件，从可视块树中筛选得到网页正文区域的候选可视块集合。如果候选可视块集合内仅有一个元素，则该元素为正文区域可视块。如果集合中存在多个候选可视块，为了明确唯一的正文区域，首先对所有的正文区域候选可视块进行遍历，删除所有存在嵌套父子关系的可视块，如图 6 所示。

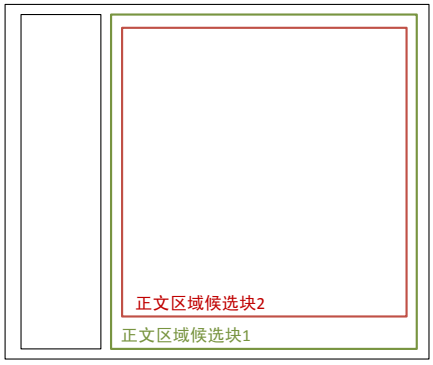


图 6 正文区域候选块的嵌套情况

Fig. 6 Nesting of main area candidate blocks

如果集合中仍旧存在多个候选可视块，则选择其中面积最大者作为正文区域可视块，其关联的数据区域即为正文区域。

5.2 局部噪声过滤

网页的数据记录对应于可视块树中的叶子节点或者子树，它们只是正文区域内的数据子块。但并不是正文区域内的所有可视块都是数据记录，在实际网页中可能存在一些不属于任何数据记录的可视块，如统计信息或者功能组件，这些块被称为是噪声块。这些噪声块为用户提供了额外的信息或快捷操作，但会对机器识别网页中有效的数据记录产生影响。如图 7 所示，噪声块 vb1（统计信息，如“总计 100 条数据”）和噪声块 vb7（功能组件，如分页跳转）分别位于正文区域的顶部和底部。需要注意的是，vb5 虽然在视觉表现上与 vb2，vb3，vb4，vb6 等数据记录相似，但是也有可能为网页的噪声块（附加信息，如广告或是网站自定义模块），这也是传统方法未曾考虑过的情况。本小节主要讲解如何去除位于正文区域首尾的噪声块，如 vb1 和 vb7。

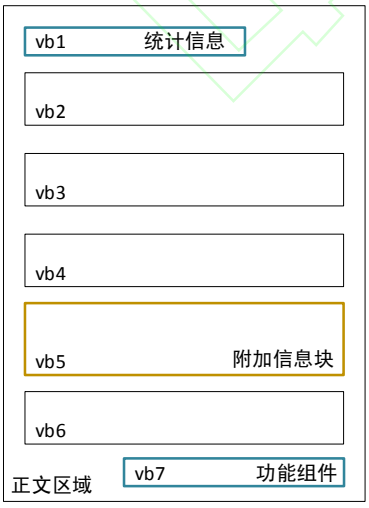


图 7 正文区域内的噪声块分布

Fig. 7 Noise block distribution in the main area

1）从位置上识别噪声块。代表数据记录的可视块往往左右边界对齐，顶部或者底部的可视块如果不满足其左右边界对齐，则被视为噪声块。

2）寻找噪声关键词，如“上一页”“下一页”。考虑正文中也有可能含有此类噪声关键词，本文进一步通过大量观察发现，噪声块中的噪声关键词通常被单独的 HTML 标签对包裹，而不是处于文本段中。因此，本文通过正则表达式匹配可视块的噪声关键词时，若该噪声关键词被单独的一对 HTML 标签包裹，则标记该可视块为噪声块。

5.3 可视块筛选

经过局部噪声过滤后，设正文区域内未被标记为噪声块的可视块集合为R。通过可视块聚类，可将集合 R 划分为多个可视块簇，并根据簇的规模判断该簇是否为网页的数据记录集合，其具体步骤如下。

首先，根据可视块的复合属性计算可视块之间的相似性。由于可视块包含由 3 种不同类型数据组成的复合信息，因此从内容方面、DOM 结构方面、视觉信息方面计算可视块的相似性是合理的。对于可视块的内容特征，本文关注可视块之间的文本长度是否接近；对于可视块的 DOM 结构特征，本文关注可视块在 DOM 树上的层级等信息；对于可视块的视觉外观特征，本文关注其字体、背景等信息。可视块 A 和可视块 B 的相似性计算公式如下：

$$Sim(a,b) = Average\big(simCONTENT(A,B) + simDOM(A,B) + simVISUAL(A,B)\big) \tag{8}$$

其中，$simCONTENT(A,B)$、，$simDOM(A,B)$、，$simVISUAL(A,B)$分别为可视块 A 和可视块 B 的内容相似性、DOM 结构相似性、视觉相似性。3 类信息的具体属性来源于前文所提的复合属性表，包含 3 大类、11 个小类别，共计 34 个不同指标，内容、DOM 结构、视觉外观方面都取各自指标的相似度平均值作为其相似性的估值。考虑 Jaccard 系数不能反映属性之间的细微差异，本文将可视块的属性分为可计数、可枚举及其他属性这 3 种情况进行量化。文本长度、DOM 深度等属性为可计数型属性，其属性值为自然数或者小数；可枚举的属性较多为与视觉相关的属性，如文本对齐方式、字体颜色等，其属性值不可计数，但是可以通过枚举获得；既无法计数，又不能枚举的属性，如可视块的文本内容，本文单独归为一类。具体的计算公式如下所示，对于可计数属性之外的属性，本文只比较 a 和 b 的值是否一致。

$$simCONTENT(A,B) = \frac{\sum_{i=1}^{n} sim(ac_i, bc_i)}{n} \tag{9}$$

$$simDOM(A,B)=\frac{\sum_{i=1}^{m} sim(ad_i,bd_i)}{m} \tag{10}$$

$$simVISUAL(A,B)=\frac{\sum_{i=1}^{x} sim(av_i,bv_i)}{x} \tag{11}$$

$$sim(a,b)=\begin{cases}1-\dfrac{(a-b)^2}{a^2+b^2} & if\ a,b\ is\ countable,\ and\ a,b\neq 0\\ 1 & if\ a,b\ is\ uncountable,and\ a=b\\ 0 & otherwise\end{cases} \tag{12}$$

为了避免聚类数量和初始点的选择过程，同时考虑时间上的消耗，VBIE 使用了密度聚类方法，基于可视块之间的相似性对集合R中的可视块进行聚类操作。其算法原理为，根据多维空间中数据点的分布，参照其密集程度，数据点被自动归并成簇。表 3 介绍了可视块聚类算法中的概念及符号，算法 1 则展示了其算法伪代码。

表 3 可视块聚类算法中的概念说明

Table 3 Conceptual description in the visual block clustering algorithm

概念名称	字母	说明
可视块集合	R	可视块数据集中所有的点形成的集合
半径参数	r	可视块簇的半径，描述为可视块与其他可形成同一个簇的可视块的最大距离
邻域	Eps	对于R中的每一个数据点，与其距离为半径参数r内的点所形成的的区域，即为该点的邻域
簇最小对象数目	$MinPts$	描述了形成可视块簇的最小可视块数目，即邻域密度阈值。当邻域中的数据点数量超过该阈值时，数据点能够在该邻域下形成一个单独的簇

算法 1 基于密度的可视块聚类

输入：：可视块集合R，半径参数r，邻域密度阈值$MinPts$： 半径参数：领域密度阈值

输出：基于密度聚类的可视块簇的集合

Step1 标记R中所有的可视块为未被访问过unvisited;
Step2 Do
Step3 随机选择一个没有访问过的可视块p;
Step4 标记该可视块p为visited;
Step5 If p以r为半径的邻域内至少有$MinPts$个可视块;
Step6 创建一个新的可视块簇C，并把 p 添加到C中;
Step7 令N为p的r邻域中的可视块集合;
Step8 For 可视块集合C中的每一个可视块p'
Step9 If p'未被访问过
Step10 标记 p'为已访问visited;
Step11 If p'不属于任何的簇，把p'加入到C中;
Step12 If p'的r邻域内至少有$MinPts$个可视块，转到Step6;
Step13 End For
Step14 输出C
Step15 Else 标记p为噪声块
Step16 Until R中所有的可视块都被访问过.

获得可视块的邻域,需要计算不同可视块之间的距离。可视块之间的距离函数公式如下所示：

$$Dis(A,B)=1-Sim(A,B) \tag{13}$$

两个可视块在视觉上越相似，则可视块之间的属性差异越小，$Sim(A,B)$趋向于 1，$Dis(A,B)$趋向于 0。

通过密度聚类，正文区域内的可视块被分至各个可视块簇中，其中最大的簇为数据记录簇，其余为噪声簇。图 8 展示了可视块聚类在必应搜索网站上的部分运行结果，可以看到所有的数据记录被凝聚为一类，而位于右下方的噪声可视块则被凝聚为其他类别。

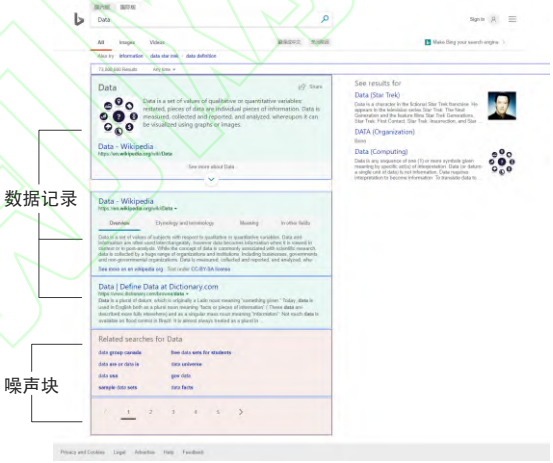


图 8 可视块聚类

Fig. 8 Visual block clustering

6 提取层：获得数据记录和数据项

数据记录为用户真正感兴趣的网页元素，其 DOM 子树对应于正文区域内的子块，而子树上的每个叶子节点对应于数据记录中包含的数据项。本文将数据记录中的数据项分为两类：1）静态的数据项，其内容不会随着数据记录的不同而改变；2）动态的数据项，其内容在不同的数据记录中都会发生变化。通常，静态数据项为网页的一些功能按钮和模板文本，如“隐藏”“感谢回复者”“转发”“点赞”等；而动态数据项才是网页信息提取关注的重点，包含作者、日期、内容等有效信息。

如图 9 所示，子图（a）为某论坛上的一段回帖列表，其中每一条评论或是回复都对应着一条该网页的数据记录；子图（b）则展示了该数据记录列表对应的数据项分布，其

中 di1，di2，di3，di4，di8 为动态数据项，在不同数据记录里具有不同的值，而 di5 和 di6 为静态数据项，其文本内容固定不变。

(a)

(b)

图 9 数据记录列表和数据项分布

Fig. 9 Data record list and data item distribution

通过结合内容信息、DOM 结构以及视觉信息，定位并挖掘数据记录中的动态数据项。其步骤大致如下：1）选择数据记录 DA，并随机选择另一个不同的数据记录 DB，提取其 DOM 子树；2）对比两棵 DOM 子树上的每一个叶子节点，如图 10 所示，如果节点的内容信息或者视觉信息发生了变化，则将其标记为动态数据项，否则为静态数据项；3）提取所有动态数据项的数据，包含其文本与视觉属性，如果动态数据项是一个锚标记，还需要提取其链接属性；4）重复以上步骤，直至所有的数据记录被访问完毕。

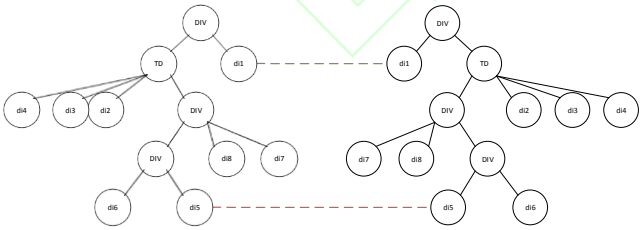


图 10 动态数据项的提取

Fig. 10 Dynamic data item extraction

7 实验

实验在 Intel(R) Core(TM) i7-7700 3.60GHz 8GB 的 PC 电脑上完成， 通过 Puppeteer 获得网页的视觉呈现，以 Chrome 浏览器作为实验平台，MongoDB 用于存储实验的结果。经由实验确定，中心偏移量比率阈值T_o、区域面积比率阈值T_a、内容长度比率阈值T_c以及区域间距比率阈值T_d分别被设置为 0.2，0.4，0.4，0.2 时算法能取得较优的结果。

实验采用了以下两个标准来评估算法的性能。

1）网页数据区域内的所有记录能否被准确地提取；

2）提取得到的数据记录中是否不包含任何不正确的数据。

设定页面内包含的有效数据记录总数为 M，提取结果的记录总数为 N，其中提取得到的正确数据记录数目为 TP，则可以得到实验结果的精确率 P、召回率 R 以及 $F1$ 值。精确率 P，即网页提取的结果中有多少是有效的数据记录；召回率 R，即应该被提取的数据记录中有多少被准确地提取；$F1$ 值，精确率和召回率的调和均值，为综合评价指标。每种衡量指标的计算方法分别如式（14）—式（16）所示。

$$P = TP/N \tag{14}$$

$$R = TP/M \tag{15}$$

$$F1 = 2PR/(P+R) \tag{16}$$

7.1 实验结果

测试集 A 采用了来自 6 个不同的主流搜索引擎网站的各 10 个搜索结果页面，包含了国内知名的百度搜索、搜狗搜索、360 搜索，以及国际上比较知名的 Google，Yahoo 及必应搜索。搜索结果页面是用户输入搜索关键词后，网页的正文部分通常由一个搜索结果的列表组成，页面布局较为单一。

如表 4 所列，对于搜索结果页面，VBIE 的网页信息提取精确率在 98%至 100%区间，$F1$ 均值为 98.5%。这些数据证明了 VBIE 在简单多记录页面的信息提取能力。VBIE 在搜狗搜索网站上召回率较低，是因为该网页的搜索记录结果中包含了较多的附加信息，如推广信息、智能推荐、智能词典等。

测试集 B 则选取了 5 个知名社区论坛网站的各 10 个页面，包含微博、豆瓣小组、V2EX、Github Community Forum 以及 Vue Forum。相比搜索结果页面，社区论坛页面的构造更为复杂，且不同网站之间的页面布局存在明显的差异。社区论坛网站的帖子页面通常会存在两种对象：话题发起者以及回复者。在大部分的社区论坛网站上，这两类信息并无差异；而在部分网站上（如微博、V2EX），话题发起者的帖子内容与回复列表存在明显的间隔，甚至两者的可视块大小上也存在较大的差异。

如表 5 所列，对于社区论坛页面，VBIE 的网页信息提取精确率和召回率略有下降，但仍旧保持在 99.5%和 93.1%左右，其最低值分别为 97.6%和 93.1%，$F1$ 的平均值为 96.1%，足以证明 VBIE 能够处理如社区论坛页面这

样的复杂多记录型网页。

由两组实验对比还可以看出，VBIE 受到网页数据记录规模以及 DOM 节点个数的影响，具有$O(nlogn)\sim O(n^2)$的时间复杂度。除此之外，VBIE 还与当前较为先进的 MDR 和 CMDR 算法进行了对比，MDR 基于 DOM 结构信息提取网页中的数据记录，CMDR 则是基于视觉信息的代表性算法之一。如表 6 所列，VBIE 在提取效果上优于 MDR 和 CMDR 方法，其提取的准确率高达 97.1%，而后两者的准确率分别是 45.5%和 80.8%，从而证明了在多记录型复杂网页的信息提取上，VBIE 是一种较为先进的有效算法，能够完成复杂多记录型网页的信息提取任务。

表 4 搜索结果网页的信息提取

Table 4 Information extraction for search-based web pages

测试集 *A*	样本数	记录数	提取数（有效数）	精确度 *P*/%	召回率 *R*/%	*F*1 值/%	平均时间消耗/s
百度	10	100	98(98)	100.0	98.0	99.0	0.090
Google	10	89	86(85)	98.8	95.5	97.1	0.147
搜狗	10	114	101(101)	100.0	88.6	94.0	0.110
必应	10	113	111(111)	100.0	98.2	99.1	0.125
360 搜索	10	96	95(95)	100.0	99.0	99.5	0.110
Yahoo	10	102	102(102)	100.0	100.0	100.0	0.113
平均值	10	102	99(99)	100.0	97.1	98.5	0.116

表 5 社区论坛网页的信息提取

Table 5 Information extraction for community forum-type web pages

测试集 B	样本数	记录数	提取数（有效数）	精确度 *P*/%	召回率 *R*/%	*F*1 值/%	平均时间消耗/s
微博	10	198	179(179)	100.0	90.4	95.0	0.292
豆瓣小组	10	597	576(576)	100.0	96.9	98.4	1.507
V2EX	10	257	254(248)	97.6	96.5	97.0	0.554
GitHub Co-mmunity Forum	10	100	90(90)	100.0	90.0	94.7	0.302
Vuejs Forum	10	154	141(141)	100.0	91.6	95.6	0.204
平均值	10	260	248(247)	99.5	93.1	96.1	0.572

表 6 多记录型页面上的算法比较

Table 6 Algorithm comparison on multi-record page

网站	样本数	帖子数	提取得到的帖子数		
			MDR	CMDR	VBIE
scam.com	10	145	80	129	140
forums.unrealengine.com	10	102	49	86	98
3dmgame.com	10	160	56	114	157
总计	30	407	185(45.5%)	329(80.8%)	395(97.1%)

结束语 本文主要提出了一种基于可视块的多记录型复杂网页信息提取算法 VBIE。首先,获取网页的视觉呈现，基于网页的复合信息构建网页的可视块和可视块树；其次,进行区域聚焦、局部噪声过滤,并筛选正文区域内的可视块；最后,提取网页的数据记录和数据项。实验表明，VBIE 框架能够对未知且复杂的多记录型动态网页进行数据提取，具有较好的泛化能力。相比其他方法，VBIE 不需要一组同源的页面样本，同时也无需特定的页面结构，利用了更多维度的网页信息，同时能够高效率、高精度、无监督地提取多记录型复杂网页中的有效信息。

在未来，VBIE 存在几个改进的方向：1）目前的数据提取建立在网页只有一个正文区域的假设下，但是实际应用中可能存在更为复杂的情况，多个难以区分彼此的数据区域会影响方法的效率；2）对于混合排版的复杂页面，部分难以分辨的噪声块仍旧可能混杂在数据记录集合中，今后可以增加多维的判断，挖掘噪声块的更多特征，提高噪声过滤和信息提取的准确率；3）效率和精度可以得到进一步的提高，如阈值的调整、算法上的改进、数据库的扩充等，空间及时间复杂度上的优化也是 VBIE 方法需要完善的地方。

参考文献

[1]中国互联网络信息中心.CNNIC 发布第 43 次《中国互联网络发展状况统计报告》 [EB/OL].(2019-02-02). http://www.cac.gov.cn/2019-02/28/c_1124175677.html.

[2]HAMMER J, MCHUGH J, GARCIA-MOLIN H. Semistructured data: the TSIMMIS experience[C]// East-European Conference on Advances in Databases and

Information Systems. British Computer Society, 1997:1-8.

[3]AROCENA G O, MENDELZON A O. WebOQL: restructuring documents, databases and Webs[C]// International Conference on Data Engineering, 1998. Proceedings. IEEE, 1998:24-33.

[4]NOVELLA T, HOLUBOVÁ I. User-Friendly and Extensible Web Data Extraction[M]//Advances in Information Systems Development. Cham:Springer,2018: 225-241.

[5]BU Z , ZHANG C , XIA Z , et al. An FAR-SW based approach for webpage information extraction[J]. Information Systems Frontiers, 2014, 16(5):771-785.

[6]OITA M, SENELLART P. FOREST: Focused object retrieval by exploiting significant tag paths[C]//Proceedings of the 18th International Workshop on Web and Databases. ACM, 2015: 55-61.

[7]SAHUGUET A, AZAVANT F. Building intelligent web applications using lightweight wrappers[J]. Data & Knowledge Engineering, 2001, 36(3): 283-316.

[8]LIU L , PU C , HAN W . XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources[C]// International Conference on Data Engineering. IEEE, 2002.

[9]BUTTLER D, LIU L, PU C. A fully automated object extraction system for the World Wide Web[C]//Proceedings 21st International Conference on Distributed Computing Systems. IEEE, 2001: 361-370.

[10]CHANG C H, HSU C N, LUI S C. Automatic information extraction from semi-structured web pages by pattern discovery[J]. Decision Support Systems, 2003, 35(1): 129-147.

[11]WEN Y, ZENG Q, DUAN H, et al. An Automatic Web Data Extraction Approach based on Path Index Trees[J]. International Journal of Performability Engineering, 2018, 14(10):2449-2460.

[12]LIU B, GROSSMAN R, ZHAI Y. Mining data records in web pages[C]//Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2003: 601-606.

[13]ZHAI Y, LIU B. Web data extraction based on partial tree alignment[C]//Proceedings of the 14th international conference on World Wide Web. ACM, 2005: 76-85.

[14]HUANG X, GAO Y, HUANG L, et al. Web Content Extraction Using Clustering with Web Structure[C] // International Symposium on Neural Networks. Cham:Springer,2017: 95-103.

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

[16]ZHAO H, MENG W, WU Z, et al. Fully automatic wrapper generation for search engines[C]//Proceedings of the 14th international conference on World Wide Web. ACM, 2005: 66-75.

[17]SIMON K, LAUSEN G. ViPER: augmenting automatic information extraction with visual perceptions[C]//Proceedings of the 14th ACM international conference on Information and knowledge management. ACM, 2005: 381-388.

[18]LIU W, MENG X, MENG W. Vide: A vision-based approach for deep web data extraction[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 22(3): 447-460.

[19]WAI F K, YONG L W, Thing V L L, et al. CMDR: Classifying nodes for mining data records with different HTML structures[C]//TENCON 2017-2017 IEEE Region 10 Conference. IEEE, 2017: 1862-1862.

[20]LIU J , LIN L , CAI Z , et al. Deep web data extraction based on visual information processing[J]. Journal of Ambient Intelligence and Humanized Computing.2017,10:1-11.

[21]GOGAR T, HUBACEK O, SEDIVY J. Deep neural networks for web page information extraction[C]//IFIP International Conference on Artificial Intelligence Applications and Innovations. Cham ： Springer,2016: 154-163.