doi:10.11959/j.issn.1000-436x.2016190

基于节点属性与正文内容的海量 Web 信息抽取方法

王海艳[1,2]，曹攀[1]

（1. 南京邮电大学计算机学院，江苏 南京 210023；2. 江苏省无线传感网高技术研究重点实验室，江苏 南京 210003）

摘 要：为解决大数据场景下从海量 Web 页面中抽取有价值的信息，提出了一种基于节点属性与正文内容的海量 Web 信息抽取方法。将 Web 页面转化为 DOM 树表示，并提出剪枝与融合算法，对 DOM 树进行简化；定义 DOM 树节点的密度和视觉属性，根据属性值对 Web 页面内容进行预处理；引入 MapReduce 计算框架，实现海量 Web 信息的并行化抽取。仿真实验结果表明，提出的海量 Web 信息抽取方法不仅具有更好的性能，还具备较好的系统可扩展性。

关键词：Web 信息；抽取；MapReduce；DOM 树

中图分类号：TP393.07 **文献标识码**：A

Information extraction from massive Web pages based on node property and text content

WANG Hai-yan[1,2], CAO Pan[1]

(1. School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China;
2. Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China)

Abstract: To address the problem of extracting valuable information from massive Web pages in big data environments, a novel information extraction method based on node property and text content for massive Web pages was put forward. Web pages were converted into a document object model (DOM) tree, and a pruning and fusion algorithm was introduced to simplify the DOM tree. For each node in the DOM tree, both density property and vision property was defined and Web pages were pretreated based on these property values. A MapReduce framework was employed to realize parallel information extraction from massive Web pages. Simulation and experimental results demonstrate that the proposed extraction method can not only achieve better performance but also have higher scalability compared with other methods.

Key words: Web information, extraction, MapReduce, DOM tree

1 引言

Proteus 工程创建者 Grishman 将信息抽取描述为“从文本中选择出的信息创建一个结构化的表现形式”[1]。作为数据挖掘的重要组成部分，信息抽取受到了众多学者的广泛关注并出现了一些相应的解决方法，如李蕾等[2]在全信息论的基础上提出的中文信息抽取系统，黄诗琳等[3]提出的从文本中抽取命名实体的方法，秦兵[4]、李天颍[5]等提出的关系信息抽取算法。近年来，随着互联网技术的普及，作为信息抽取的重要分支，Web 信息抽取技术也得到了极大的发展，出现了诸多的抽取算法，主要有如下几类。

基于视觉分块抽取方法最早由微软亚洲研究院的 Cai 等[6]提出的，该方法主要通过页面分块的视觉特征量对页面内容进行分类来抽取正文信息，

收稿日期：2015-11-16；**修回日期**：2016-05-24

基金项目：国家自然科学基金资助项目（No.61201163, No.61672297）；“六大人才高峰”基金资助项目（No.2013-JY-022）；江苏省“333 高层次人才培养工程”基金资助项目

Foundation Items: The National Natural Science Foundation of China (No.61201163, No.61672297), Six Talent Peaks Project in Jiangsu Province (No.2013-JY-022), 333 High Level Personnel Training Project in Jiangsu Province

在此基础上，Neil[7]提出了 Extractor 算法；Narwal 等[8]提出了基于页面内容布局的信息抽取方法。相对于基于视觉分块的方法，基于 DOM 树的更容易实现，如 Sun 等[9]在根据 DOM 树节点的文本密度来实现 Web 信息的抽取；张乃洲等[10]将 DOM 树技术与标签传播相结合构建了统一的信息抽取框架；Wan 等[11]在 DOM 树的基础上引入朴素贝叶斯的文本分类模型，实现中文网站的信息抽取。除了上述 2 种方法之外，还有基于模板的抽取技术，这种算法主要抽取的对象为通过固定模板生成的网页，如 Krishn 等[12]提出的算法等。

但是，已有的这些方法要么算法的复杂度比较高，信息抽取的效率较低，要么页面切割的思想较为简单，不能够较好地对页面中的各类数据实现完整的抽取。另外，由于现有的大多数方法都是在单线程的基础上进行线性抽取，显然无法满足当前大数据环境下信息抽取的需求，而 MapReduce 作为一个高度并行的编程模型，近年来，已经运用到很多领域，如 Mansurul 等[13]提出的基于 MapReduce 的频繁子图挖掘算法和 Jin 等[14]提出了并行空间协同定位算法。这些研究都表明 MapReduce 作为一个高度统一的编程模型在处理海量数据时是高效且可行的。因此，基于上述分析，本文在现有理论的基础上，针对当前大数据环境下 Web 信息抽取存在的准确率和召回率不高以及抽取效率较低问题，提出一种基于节点属性与正文内容的海量 Web 信息抽取方法，主要工作如下。

1) 提出针对 DOM 树的剪枝与融合方法，在保证信息不丢失的基础上，有效简化 DOM 树，提高后续信息抽取的准确率和效率。

2) 充分考虑 DOM 树节点的密度和视觉属性，根据属性值对 Web 页面内容进行有效的预处理。

3) 引入 MapReduce 计算框架，实现海量 Web 信息的并行化抽取，提高信息抽取的效率。

2 相关概念

2.1 HTML DOM

文档对象模型（DOM, document object model），是 W3C 组织推荐的处理可扩展标志语言的编程接口，它可以用来访问和处理 HTML 文档，并将其转换为一个节点树形结构，根据 W3C 的 HTML DOM 标准，HTML 文档中的所有内容都是节点。下述为一段标准的 HTML 代码。

```
<HTML>
    <head>
        <title>document title</title>
    </head>
    <body>
        <a>my link</a>
        <h1>my title</h1>
    </body>
</HTML>
```

其对应的 DOM 树结构如图 1 所示。

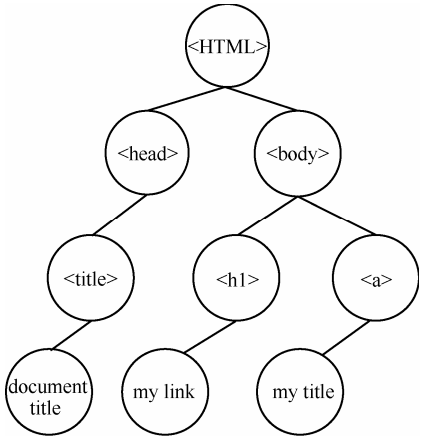

图 1 HTML 对应的 DOM 树结构

2.2 朴素贝叶斯定理

贝叶斯定理的核心思想是通过对某一个事件过去发生的概率情况来推断当前这一事件发生的概率的计算方法。贝叶斯分类法基于贝叶斯定理，其在文本分类中已被广泛使用，它们可以预测类隶属关系的概率，如一个给定元组属于一个特定类的概率。

设 X 是数据元组，通常由 n 个属性集的测量值描述。令 H 为某种假设，如数据元组 X 属于某个特定类 C。$P(H|X)$是后验概率，即在条件 X 下，H 的后验概率，$P(H)$是先验概率，即 H 的先验概率。类似地，$P(X|H)$是条件 H 下，X 的后验概率，$P(X)$是 X 的先验概率。$P(X)$、$P(H)$、$P(X|H)$可以由给定的数据估计，贝叶斯定理通过这三者概率值计算后验概率 $P(H|X)$，计算式为

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)} \tag{1}$$

朴素贝叶斯分类法是在贝叶斯分类法的基础上假定一个属性值在给定类上的影响独立于其他属性的值，分类原理为对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，通过概

率值判定待分的类归属的类别。通过这样的假设，极大地简化了计算，并通过相关实验论证其具备较高的准确率。

3　海量 Web 信息抽取方法

3.1　基于 MapReduce 的抽取框架

为了提高在海量数据环境下 Web 信息抽取的效率，引入 MapReduce 计算框架，实现抽取的并行化处理。首先，将本地待处理的网页存储于 HDFS 上，HDFS 将这些数据动态的分配给 Hadoop 平台中的每个 Map 节点，为了让每一个节点都能完整且并行的处理数据，本文将抽取算法完整的映射到每一个节点上。Map 节点在每一次 Map 过程中都将对网页进行完整的信息抽取，最终每个节点都将产生完整的信息，Reduce 函数将以键值对的形式将这些数据输出。下面将具体介绍 Map 以及 Reduce 过程。

Map 过程将主要包含 Web 页面向 DOM 树的转换、DOM 树的简化、基于节点属性的 Web 信息预处理以及在预处理基础上根据节点内容实现的信息抽取等 4 个模块，具体的函数如下。

算法 1　Map 过程

输入　数据格式为<key, value>，具体函数中为<pageNumber, webpage>, pageNumber 为待抽取页面的序列号，webpage 为待抽取的页面。

输出　数据格式为<key, value>，具体函数中为<pageNumber, content>，pageNumber 为经过抽取后的页面序列号，而 content 则是从页面中抽取出来的有价值内容。

1) Procedure Map（offset key,object value）

2)　for all webpages in List webpages do{

3)　extraction Model model = createExtractionModel(webpage);

4)　DOMTree tree = model.transfer (webpage); //页面转化为 DOM 树

5)　SimpleTree st = model.simplify (tree); //DOM 的剪枝融合操作

6)　st = model.Pretreatment(st); //基于节点属性 Web 信息预处理

7)　st = model.Extraction(st); //在预处理基础上根据节点内容完成抽取工作

8)　emit (key, webpage);//输出页面中有价值的内容块

9)　}

Reduce 过程是对 Map 过程的输出结果进行合并处理，每个 Reduce 过程都对应一个独立的输出文件。在 Reduce 过程之前，系统将执行一个 Sort 和 Partition 过程，Sort 会将 Map 的输出结果按照 pageNumber 值进行自然排序，Partition 将对排序的结果进行划分，某一个范围内的数据将会被分配到同一个 Reduce 函数中，极大地简化了任务的处理，有效地提高了效率。经过 Reduce 函数处理之后，将得到待抽取页面中有价值内容块的合集存储到 HDFS 中，完成 Web 信息抽取整个操作。

3.2　DOM 树的剪枝融合

在本文采用开源项目 HTMLParser 将 Web 页面转化为 DOM 树结构，与其他解析器相比，HTMLParser 具有较好的解析功能，相关的实验表明它能够正确、快速地解析大部分的 HTML 文档。然而直接对 Web 页面转换后的 DOM 树进行信息抽取会在一定程度上降低方法的准确率和效率，因为 Web 页面中存在着许多无意义的内容，如脚本语言、版本信息以及其他类型的标签等，根据常规的 Web 页面编码以及 W3C 相关规定，这些标签通常不包含正文内容。为了提高信息抽取的效率和准确率，本文对这些元素进行剪枝操作，删除内容如下。

1) head 标签及其包含的内容，包括网站的标题信息和页面的样式链接。

2) script 标签及其包含的内容，包括一些内嵌的样式链接和样式文本。

3) 类型为 hidden 的标签及其包含的内容，包括一些隐藏的导航信息、注册信息等。

4) form 标签及其包含的内容，包括一些搜索模块、登录模块和注册模块等。

5) HTML 文档中的注释信息、 、命名空间等内容。

6) 内容为空的标签元素。

7) 其他一些自定义的非常规元素。

考虑到基于以上规则进行剪枝操作之后的 DOM 树中仍然存在大量的节点，如
、<p>、<tr>、<td>等标签节点，这些标签节点会让语义上本属于同一个内容块的文本被分割成 2 块，因此，必须对这些节点进行融合，这样有效避免了由于信息量过短而导致的抽取精度下降等问题。根据 W3C 标准，DOM 树中的节点可分为 2 类：块元素和内联元素，块元素一般作为容器出现，用来组织结构，它们可以包含内联元素，正文内容以及其他块元素，如

<h1>、<p>、 等，通常在页面编码的时候，块元素作为同一语义块的最大容器，这样有效避免了由于信息量过短而导致的抽取精度下降等问题；内联元素只能包含文本内容和内联元素，如<a>、、等。基于这样的标准，本文提出针对 DOM 树的融合方法，融合规则如下。

1) 若内联元素有子节点，则将其子节点与其合并，此子节点也包括属性节点，并标记属性节点的特征。

2) 若内联元素为某块元素的子节点，且该块元素有其他子节点，则将其与块元素的其他子节点合并。

3) 存在兄弟关系或子孙关系的块元素均不合并，保证页面中的每一块独立的信息都对应 DOM 树中的一个节点。

根据上述定义的剪枝规则以及融合规则给出对应的算法。

算法 2 剪枝融合算法

输入：DOM 树根节点

输出：ST 树

1) Procedure Simplify (root)

2) for each child node in DOMTree do{

3) if node in delete //delete 中的元素为剪枝规则中的元素

4) remove node from DOMTree , continue;

5) else if curnode in fusion1 { //fusion1 中的元素为融合规则 1 中定义的元素

6) node = node+node's childnode;

7) remove node's childnode from DOMTree , continue;

8) }

9) else if node in fusion2 { //fusion2 中的元素为融合规则 2 中定义的元素

10) parentnode's child = node+ parentnode's otherchild;

11) remove node and parentnode's otherchild from DOMTree , continue;

12) }

13) else if node in fusion3 //fusion3 中的元素为融合规则 3 中定义的元素

14) continue the next Simplify(node);

15) }

图 2 给出了调用算法 1 对 DOM 树进行剪枝融合的示意，虚线框内的元素为待融合的元素，并将简化后的 DOM 树定义为简单树 ST。

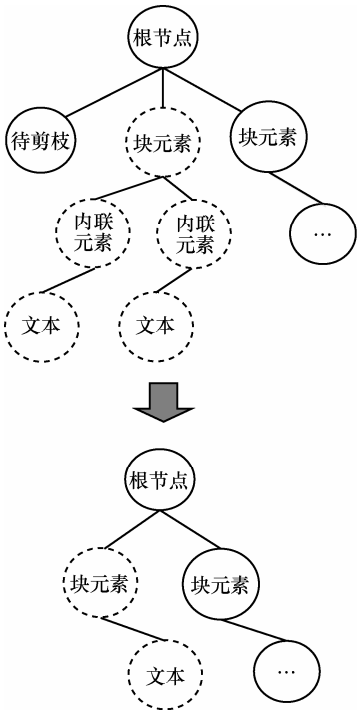

图 2 剪枝融合示意

3.3 正文内容抽取

正文内容抽取是为了有效地对页面中不同区域块中的信息进行识别以消除噪音源，从而达到抽取正文信息的目的。传统的基于统计的信息抽取方法所定义的密度属性要么过于复杂，通常包含多个属性值，导致最终抽取过程中时间的消耗较多；要么过于简单，仅仅从单一的文本密度角度出发，这在传统的单正文体中是比较合适的，但在 UGC 类型的页面中则很难达到一个令人满意的结果，综合考虑当前的现状，本文从纯文本密度和链接文本密度角度出发，首先页面中信息进行如下分类。

1) 纯文本信息，包括正文信息、网站的版权信息以及相关地址信息等。

2) 链接信息，本文中的链接信息为广义上的链接，其不仅包括网站的功能性链接、广告链接以及正文区域中相关链接，还包括网站中的各类图片链接（对于图片链接的分析，将所有的相对地址转化为绝对地址）。

在上述分类的基础上，统计 ST 树中纯文本数量以及每个节点的纯文本数量和链接数量，分别记为 *text*、*node.text* 和 *node.link*，并对 ST 树节点的密度属性做出如下定义。

定义 1 纯文本密度，其为 *node.text* 与 *text* 的比值。

$$\rho_1 = \frac{node.text}{text} \tag{2}$$

它主要表示在 ST 中，各个节点所包含的纯文本量占总文本量的比重。通过观察发现，若网页的正文区域是以大量文本构成时，节点值越大，其越有可能包含正文信息内容。

定义 2　链接文本密度，其为 *node.link* 与 *node.text* 的比值。

$$\rho_2 = \frac{node.link}{node.text} \tag{3}$$

它主要表示在 ST 中，各个节点中链接数量与纯文本数量的比值。通常正文区域的链接前后包含一定量的非链接文本而非正文区域的链接所在的节点通常不具备这样的特征，它一般不含任何形式的文本或只包含链接文本。

此外，考虑到通常单一的基于节点密度属性的 Web 信息抽取方法的仍然难以达到一个令人满意的准确率，本文在节点密度的基础上本文定义了内容块的视觉相关属性。因为在一般的网页中有价值的内容会集中于网页中心位置，而其他信息，如广告信息、导航信息、网站的版权信息一般会放置于网页的边缘，基于这样的特征，可以通过视觉属性较好的过滤掉网页中处于边缘位置的无价值信息。下面给出 ST 节点的视觉属性定义，具体的属性值通过 CSS 文件读取。

定义 3　视觉特征量<*top*，*bottom*，*width*，*height*>。该四元组中，*top* 表示页面分块与页面顶端的距离，*bottom* 表示页面分块与页面底端的距离，*width* 表示页面分块的宽度，*height* 表示页面分块的高度，将这些数值映射到区间[0, 1]，当 *top* 值大于 *bottom* 值时，采用<*bottom*，*width*，*height*>三元组来衡量内容块的重要性，反之则采用<*top*，*width*，*height*>三元组来衡量内容块的重要性。

根据上述定义的节点密度和视觉特征量以及相关抽取规则给出对应的算法。

算法 3　正文内容抽取算法

输入：ST 树

输出：页面正文区域内容

```
1) Procedure Extraction(ST)
2) get(ST's leafnode);
3)     for each leafnode in ST do{
4)      if top>bottom{
5)      if <bottom, width, height> is not in Δ3
//Δ3 为节点视觉特征量阈值集合
6)              remove leafnode from ST;
7)          else
8)              density (leafnode);
9)       }
10)     else {
11)       if <top, width, height> is not in Δ4
//Δ4 为节点视觉特征量阈值集合
12)           remove leafnode from ST;
13)       else
14)           density (leafnode);
15)      }
16)     function density(leafnode){
17)       if  ρ1<Δ1          //Δ1 表示纯文本密
度阈值
18)           remove leafnode from ST;
19)       else  if ρ2>Δ2          //Δ2 表示链接文
本密度阈值
20)           remove leafnode from ST;
21)      }
22)}
```

3.4　基于节点内容特征的 Web 信息抽取

经过上述的预处理操作，ST 中的节点将只包含正文内容，由于正文内容的主体也有可能由一些广告内容、灌水内容以及其他一些价值率较低的内容构成，因此在下一步的工作中需要对这些信息进行有效的分类，以提高信息的价值率。本文对正文中的内容给出如下分类。

第 1 类：网页的主题由广告内容，灌水内容以及其他一些价值率较低的内容构成，对于这样的页面进行整体过滤。

第 2 类：网页主题的内容具有一定的抽取价值，但是主题下存在一些价值率较低的回复，在保留主题基础上，对这些回复进行过滤。

基于上述分类，在预处理的基础上引用基于朴素贝叶斯的文本分类模型来完成最终的抽取工作，并选择停用词表和特征项选择的方法对正文内容进行有效的降维。

1) 停用词表：正文中的连接词、代词等对内容分类不仅没有作用，还会起到干扰，所以本文建立

一个停用词表，将这类词存放其中，若正文中出现停用词表中的词则将其删除。

2) 特征项选择：从高维度向量空间中选取对文本分类有效的特征项 X_i，从而达到对向量空间降维的目的，本文采用信息增益的方法进行特征项选择。

将 ST 树中节点包含的文本信息记为 d，由 n 个属性的测量值表示，其特征向量为 $\boldsymbol{T}=\{x_1,x_2,x_3,\cdots,x_{n-1},x_n\}$，$x_i\in\{0,1\}$，$x_i$ 为 d 的特征项 X_i 的特征值，若 $x_i=1$，表示其特征项在 d 中出现，$x_i=0$ 则表示特征项没有在 d 中出现，根据式(1)，可知所选取的节点属于 c_j($c_j\in C$，C 为正文内容的类别)的概率为

$$P(c_j\mid d)=\frac{P(\boldsymbol{X}=\boldsymbol{T}\mid c_j)P(c_j)}{P(\boldsymbol{X}=\boldsymbol{T})},j\in\{0,1\} \tag{4}$$

由全概率公式得

$$P(\boldsymbol{X}=\boldsymbol{T})=\sum_{j=0}^{1}P(c_j)P(\boldsymbol{X}=\boldsymbol{T}\mid c_j) \tag{5}$$

大量的文本信息将导致构成向量$\boldsymbol{X}$的特征值非常多，且特征值之间可能存在一定的关联，直接由上述公式计算出 d 的归属概率是非常复杂的，因此，在贝叶斯定理的基础上采用朴素贝叶斯定理，假定各属性之间的相互影响是独立的，即从 d 中提取出来的词之间是没有关联的，通过这种假设，朴素贝叶斯模型不仅表现出高速度，并且也有较高的准确率，则由此假设可得

$$P(\boldsymbol{X}=\boldsymbol{T}\mid c_j)=\prod_{i=1}^{n}P(X_i=x_i\mid c_j),j\in\{0,1\} \tag{6}$$

由上述公式，可推演出 d 属于某一类的概率的朴素贝叶斯公式为

$$P(c_j\mid d)=p(c_j\mid\boldsymbol{X}=\boldsymbol{T})$$

$$=\frac{P(c_j)\prod_{i=1}^{n}P(X_i=x_i\mid c_j)}{\sum_{j=0}^{1}P(c_j)\prod_{i=1}^{n}P(X_i=x_i\mid c_j)},\ j\in\{0,1\} \tag{7}$$

根据式(7)，可以推算出内容为无价值的概率为 $P(c_0|d)$，内容为有价值的概率为 $P(c_1|d)$，若 $P(c_0|d)>P(c_1|d)$，则判定当前节点包含的内容为无价值的。

若 ST 树中的首个节点包含的内容被判定为无价值，则表示当前抽取的网页属于第 1 种类型，停止对当前 Web 页面的抽取工作，将网页从本地删除，若非首节点内容被判定为无价值内容，则表示当前抽取的网页属于第 2 种类型，删除当前被判定为无价值内容的节点。

3.5 信息抽取流程

基于上述提出的相关算法与框架，下面给出基于节点属性与正文内容的海量 Web 信息抽取方法完整流程。

步骤 1：将本地数据库中 Web 页面加载到 HDFS 中。

步骤 2：通过 HDFS 将数据分配给每个 Map 节点，并将完整的抽取算法映射到 Map 节点。

步骤 3：调用 HTMLParser 将网页转化为 DOM 树。

步骤 4：调用剪枝融合算法对 DOM 树进行简化操作，形成简单树 ST。

步骤 5：根据 DOM 树节点属性对 ST 树中的节点进行筛选分类，实现 Web 信息的预处理。

步骤 6：引入基于朴素贝叶斯的文本分类模型，根据节点内容，在预处理的基础上实现完成信息最终的抽取工作。

步骤 7：Reduce 函数将抽取出的内容块以键值对的形式输出，存储于本地数据库中。

4 实验结果与分析

为体现本文提出的信息抽取方法的性能（准确率和运行效率），将实验分为 2 部分：实验 1 将本文提出的方法与 Sun 等[9]中提出的算法以及 Wang 等[11]提出的算法进行对比，他们的算法分别从是从 DOM 树的角度以及贝叶斯的角度进行信息的抽取，因此，可以通过对比 3 种方法各自的 Web 信息抽取结果来验证算法的准确率；实验 2 将验证本文提出的基于 MapReduce 的海量 Web 信息抽取的可行性，分别从数据的扩展性和机器的扩展性 2 个角度，通过计算相应的加速比和时间消耗来评估本文方法的性能。

4.1 数据集

为了达到较好的实验效果，本文以爬虫的方式从互联网上收集约 10 000 张页面的数据量，分别来自百度贴吧、CSDN、小木虫等主流站点，页面内容涵盖了社会、计算机编码、学术科研等多种类型，这些网站的页面结构与风格迥异，有助于充分地验证本文方法的准确性、高效性以及顽健性。实验 1 是为了验证 3 种方法对不同结构的页面的抽取效果，将从 10 000 张页面中随机挑选出 240 个页面进行相关的抽取工作。实验 2 首先验证本文方法的机

器可扩展性，在具体的实验环境中将系统的节点数分为设为 1、2、4、6、8 个，抽取对象均为当前加载到本地的 10 000 张页面，其次验证本文方法的数据可扩展性，具体的实验环境中设立 8 个节点，处理的页面量分别为 100、500、2 000、5 000、8 000 张复杂度相差无几的页面。

4.2　评价标准

对于 DOM 树的简化过程，本文采用平均融合率 *fusionRate* 来验证此过程的必要性，其计算式如下

$$fusionRate = \frac{\sum_{i=0}^{n} STNum_i}{\sum_{i=0}^{N} DOMNum_i} \tag{8}$$

其中，*STNum* 为 DOM 树中被剪枝融合掉的节点数量，*DOMNum* 为剪枝融合之前 DOM 树的节点数量。此外，每一个 DOM 树可能对应于多个简单树，因此，本文将剪枝融合得到的简单树数量定义为 *n*，原始的 DOM 树数量定义为 *N*。

在 Web 信息的预处理实验中，采用准确率（Precision）、召回率（Recall）以及 F-Score 这 3 种指标来评价实验结果。准确率的计算式为

$$P = \frac{TP}{TP+FP} \tag{9}$$

TP 表示抽取出的信息中有效信息量，*FP* 表示抽取出的信息中包含的无效信息量。召回率计算式为

$$R = \frac{TP}{TP+FN} \tag{10}$$

TP 为抽取出的信息中有效信息量，*FN* 表示未被抽取出的信息中有效信息量。F-Score 计算式为

$$F_1 = \frac{2PR}{P+R} \tag{11}$$

其中，*P* 和 *R* 分别表示为信息抽取的准确率和召回率。

4.3　结果与分析

实验 1　Web 信息抽取准确率验证

表 1 给出了实验过程中 DOM 在简化前后节点数量的变化情况，随着实验数据量的递增，DOM 树的平均节点数基本维持在 1 700 个左右，而经过处理后的 DOM 树只有大约 600 个节点，平均融合率达到了 62.8%，极大简化了后续信息抽取工作，一定程度上提高了方法的执行效率，充分说明了预处理阶段对 DOM 树进行剪枝融合的必要性与有效性。

表 1　**DOM 树的剪枝融合结果**

页面数量	每页平均节点数		融合率	平均融合率
	处理前	处理后		
30	1 653	603	63.5%	62.8%
60	1 820	720	60.4%	
90	1 673	659	60.6%	
120	1 683	647	61.6%	
150	1 745	635	63.6%	
180	1 670	603	63.9%	
210	1 735	620	64.3%	
240	1 698	597	64.8%	

此外，从实验 1 中的 240 张页面中随机地抽取 50 张页面进行正文内容损耗统计，统计结果表明本文所提出的剪枝融合方法不会对损害正文区域的有效内容，并且也不会对隶属于不同语义的内容进行错误融合。

表 2 给出了本文算法、Sun 等[10]提出的算法以及 Wang 等[11]提出的算法抽取结果，从数据中可以看到，3 类信息抽取方法的召回率基本相近，但是在准确率上，本文提出的方法要明显优于其他 2 种方法，并且得到了最高的 F_1 值，主要原因如下。

1) 设计了剪枝融合算法，对 DOM 树进行了简化操作，过滤了页面中大量无效信息块，并且对语义上属于同一个内容块的信息进行了融合，有效降低了信息量较少的内容块对信息抽取造成的干扰。

2) 将 DOM 树节点的密度与视觉属性相结合，可完成对页面中链接、图片、文本等各种信息的抽取，并且能对正文区域中发布者的身份、头像、个性签名等无效信息进行有效的过滤。

3) 引入基于朴素贝叶斯的文本分类模型，对正文区域中的无价值内容进行了分类过滤，有效地提高了信息的价值率。

实验 2　基于 MapReduce 的可行性验证

为了观察方法的并行化处理框架在集群规模增大时其性能变化情况，进行了系统的可扩展性实验。在实验中采用本地的 10 000 张 Web 页面进行抽取工作，具体的集群中将分别采用 1、2、4、6、8 个计算节点，实验结果数据处理后如表 3 所示。

表 2 Web 信息抽取结果

页面数量	本文方法			文献[10]方法			文献[11]方法		
	P	*R*	F_1	*P*	*R*	F_1	*P*	*R*	F_1
30	**0.89**	0.94	**0.91**	0.83	0.94	0.88	0.78	**0.96**	0.86
60	**0.92**	0.95	**0.93**	0.84	**0.96**	0.90	0.77	**0.96**	0.85
90	**0.90**	**0.96**	**0.92**	0.84	**0.96**	0.89	0.80	0.95	0.87
120	**0.91**	0.96	**0.93**	0.82	0.95	0.88	0.76	**0.97**	0.85
150	**0.90**	**0.95**	**0.92**	0.80	**0.95**	0.87	0.75	**0.95**	0.84
180	**0.93**	**0.96**	**0.95**	0.81	**0.96**	0.88	0.75	**0.96**	0.84
210	**0.92**	**0.95**	**0.93**	0.81	**0.95**	0.87	0.76	**0.95**	0.84
240	**0.91**	**0.96**	**0.93**	0.81	0.95	0.87	0.76	**0.96**	0.85

在表 3 中设立了总时间消耗和平均时间消耗，并通过这 2 个度量属性计算出相应的加速比，将其作为衡量本文的分布式系统可扩展性的重要指标。从具体的数据中可以看到在前 6 个节点之前随着系统环境中计算节点的增加其对应的加速比几乎是呈线性化增长的，但超过 6 个节点以后其对应的加速比逐渐放缓，这主要是因为当节点数增加整个算法的并行化的额外开销也增加，在当前数据量较小的情况下，并不能较好地体现多节点高度并行的处理优势。

但总体上来说，表 3 中的数据可以说明随着处理器数量的增长，系统对页面处理的时间下降以及加速比的增长都十分显著，在一定程度上说明了本文提出的方法具备较好的系统可扩展性。

表 3 系统可扩展性验证

节点数	总时间/s	平均时间/s	加速比
1	46 875	4.69	1
2	23 721	2.36	1.99
4	12 460	1.20	3.96
6	8 343	0.83	5.85
8	6 154	0.62	7.78

在对算法的系统可扩展性进行了相关验证的同时，本文对方法的数据可扩展性也进行了相关的实验，在具体的实验环境中，通过人工筛选的方式尽可能地选用复杂度相同但数据规模不同的 5 组数据，数据的规模从 100 张页面到 8 000 张页面，对不同规模的数据分别做 2 组实验，取其平均值作为最终的耗时。

在表 4 中设立了总时间消耗和平均时间消耗并将其作为系统数据可扩展性的评价指标。从具体的数据中可以看到在处理页面的数量从 100 张增加到 2 000 张的过程中，页面的平均处理时间有一个减少的过程，在 2 000 张页面之后，页面的平均处理时间趋向一个稳定的值，这主要是因为在数据量较少时，算法的并行化的额外开销在总体耗时的比重过大，其在一定程度上影响了抽取的效率。

但总体上来说，表 4 中的数据在一定程度上可以较好地说明本文提出的方法在面对页面数量不断增长时能够表现出较好的抽取性能，有效地验证了方法的数据可扩展性。

表 4 数据可扩展性验证

节点数	页面量	总时间/s	平均时间/s
8	100	65	0.65
	500	318	0.64
	2 000	1 265	0.63
	5 000	3 157	0.63
	8 000	5 044	0.63

5 结束语

如何有效地抽取 Web 页面中有价值的内容是当前 Web 研究领域的热点问题，本文根据现有 Web 信息抽取方法在大数据环境下的不足，提出一种基于节点属性与内容的海量 Web 信息并行抽取方法。该方法根据当前 Web 前端编码的特点以及 W3C 的相关规定，充分考虑节点的视觉与密度属性以及节点包含的内容的特点，可完全独立于 Web 页面的结构，相关实验表明了本文的方法是高效且可扩展的。

在未来的工作中，将对本文提出的方法进行深入研究，进一步优化 MapReduce 抽取框架，提高系

统的性能。

参考文献：

[1] GRISHMAN R. Information extraction: techniques and challenges［EB/OL］. http:// cs.nyu.edu/cs/faculty/grishman/proteus.htm, 1997.

[2] 李蕾, 周延泉, 王菁华.基于全信息的中文信息抽取系统及应用[J]. 北京邮电大学学报, 2005, 28(6): 48-51.
LI L, ZHOU Y Q, WANG J H. Comprehensive information based chinese information extraction system and application[J]. Journal of Beijing University of Posts and Telecommunications, 2005, 28(6): 48-51.

[3] 黄诗琳, 郑小琳, 陈德人.针对产品命名实体识别的半监督学习方法[J]. 北京邮电大学学报, 2013, 36(2): 20-23.
HUANG S L, ZHENG X L, CHEN D R. A semi-supervised learning method for product named entity recognition[J]. Journal of Beijing University of Posts and Telecommunications, 2013, 36(2): 20-23.

[4] 秦兵, 刘安安, 刘挺. 无指导的中文开放式实体关系抽取[J]. 计算机研究与发展, 2015, 52(5):1029-1035.
QIN B, LIU A A, LIU T. Unsupervised Chinese open entity relation extraction[J]. Journal of Computer Research and Development, 2015,52(5):1029-1035.

[5] 李天颍, 刘璘, 赵德旺, 等. 一种基于依存文法的需求文本策略依赖关系抽取方法[J]. 计算机学报, 2013,31(1):54-62.
LI T Y, LIU L, ZHAO D W, et al. Eliciting relations from requirements text based on dependency analysis[J]. Journal of Computers, 2013, 31(1): 54-62.

[6] DENG C, YU S P, WEN J R. VIPS: a vision-based page segmentation[R]//Microsoft Technical Report, MSR-TR_ 203-79,2003.

[7] NEIL A, HONG J. Visually extracting data records from the deep-Web[C]//WWW 2013. Rio, IEEE Press, 2013: 1233-1238.

[8] NARWAL N. Improving Web data extraction by noise removal[C]// ARTCom 2013. Bangalore, IET, 2013: 388-395.

[9] SUN F, SONG D, LIAO L. DOM based content extraction via text density[C]//ACM SIGIR 2011. Beijing, 2011: 245-254.

[10] 张乃洲, 曹薇, 李石君. 一种基于节点密度分割和标签传播的 Web 页面挖掘方法[J]. 计算机学报, 2015, 38(2): 349-364.
ZHANG N Z, CAO W, LI S J. A method based on node density segmentation and label propagation for mining Web page[J]. Journal of Computers, 2015, 38(2): 349-364.

[11] WANG J B, WANG L Z, GAO W L, et al. Chinese Web content extraction based on naive bayes model[C]// International Federation for Information Processing IFIP. 2014: 404-413.

[12] KRISHNA S S, DATTATRAYA J S. Schema inference and data extraction from templatized Web pages[C]//ICPC, 2015: 1-6.

[13] BHUIYAN M A, ALHASAN M. FSM-H: frequent subgraph mining algorithm in Hadoop[C]//Big Data. 2014: 9-16.

[14] JIN S Y, BOULWARE D, KIMMEY D. A parallel spatial co-location mining algorithm based on MapReduce[C]//Big Data. 2014: 25-31.

作者简介：

王海艳（1974-），女，江苏东台人，南京邮电大学教授，主要研究方向为服务计算、可信计算、大数据应用与云计算技术、隐私保护技术。

曹攀（1991-），男，江苏镇江人，南京邮电大学硕士生，主要研究方向为云计算与物联网技术。