计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210100230

基于特征相似度计算的网页包装器自适应

陈迎仁　郭莹楠　郭　享　倪一涛　陈　星
福州大学数学与计算机科学学院　福州 350108
福建省网络计算与智能信息处理重点实验室(福州大学)　福州 350108
(2318191704@qq.com)

摘　要　随着大数据的发展,互联网数据呈现爆炸式的增长。Web作为一种重要的信息载体,包含了各种类型的信息,而包装器的提出就是为了从杂乱的Web信息中提取出目标数据。但是,随着网页更新的频繁,轻微的结构变化都可能导致原有的包装器失效,增加包装器的维护成本。针对包装器的健壮性以及维护成本问题,提出了一种基于特征相似度计算的网页包装器自适应技术。该技术主要通过解析新网页的特征集合和旧包装器所蕴含的特征信息,通过网页相似度计算,重定位旧包装器在新网页中的映射区域和映射数据项,并根据映射关系使旧包装器能够自适应新网页的数据提取。该技术主要针对各类型网站进行实验,其中包括了购物类、新闻类、资讯类、论坛类和服务类,从中选取了250对新旧版本网页,共500个网页,进行包装器自适应实验。实验结果表明,当网页结构改变时,该方法能够有效地自适应新网页的数据提取,且数据提取的平均精确度和平均召回值分别达到82.2%和84.36%。

关键词:网页数据抽取;自适应;包装器;相似度计算;网页特征
中图法分类号　TP311

Web Page Wrapper Adaptation Based on Feature Similarity Calculation

CHEN Ying-ren,GUO Ying-nan,GUO Xiang,NI Yi-tao and CHEN Xing
College of Mathematics and Computer Science,Fuzhou University,Fuzhou 350108,China
Fujian Key Laboratory of Network Computing and Intelligent Information Processing (Fuzhou University),Fuzhou 350108,China

Abstract　With the development of big data,Internet data has exploded. As an important information carrier,the Web contains various types of information. The wrapper is proposed to extract target data from messy Web information. However,with frequent Web page updates,minor structural changes may cause the original wrapper to fail,leading to increased maintenance costs for the wrapper. Aiming at the robustness and maintenance cost of the wrapper,a Web page wrapper adaptive technology based on feature similarity calculation is proposed. This technology mainly analyzes the feature set of the new Web page and the feature information contained in the old wrapper,and calculates the similarity of the Web page to relocate the mapping area and mapping data items of the old wrapper in the new Web page,and make the old wrapper based on the mapping relationship able to adapt the data extraction of new Web pages. The technology is mainly used for experiments on various types of Websites,including shopping,news,information,forums and services. 250 pairs of old and new versions of Web pages,totaling 500 Web pages,are selected for wrapper adaptation experiments. The experimental results show that when the Web page structure changes,the method can effectively adapt to the data extraction of the new Web page,and the average precision and average recall of data extraction reach 82.2% and 84.36%,respectively.

Keywords　Web page data extraction,Adaptation,Wrapper,Similarity calculation,Page features

1　引言

随着互联网的不断发展,互联网数据呈现出爆炸式的增长,其中包含了各式各样的数据[1]。Web作为信息的重要载体,目前其数量已经达到千亿级别,而且网页结构往往是复杂的。各种数据存在于复杂的网页结构之中,其中还包含了许多噪音信息,而Web数据抽取技术[2]的出现就是为了从无结构化或半结构化的Web页面中抽取出用户所需的目标数据,并以结构化、语义清晰的格式存储[3]。于是诞生了包装器(Wrapper)的概念,一个能够将数据从HTML网页中抽取出来,并且将它们还原为结构化的数据的软件程序[4]。

但是通过传统的网页包装器进行Web数据抽取仍然存在以下问题:在网页的生命周期中,网页的更新迭代频繁,而且许多的网页更新只是轻微的结构变化,但网页结构的轻微变化却可能导致原有的包装器无法正常工作[5]。这些问题的存在,使得包装器的维护需要较大的成本投入,这就对包装器

基金项目:国家重点研发计划(2017YFB1002000);福建省自然科学基金杰青项目(2020J06014);福建省自然科学基金项目(2018J07005)
This work was supported by the National Key R&D Program of China(2017YFB1002000),Natural Science Foundation of Fujian Province for Distinguished Young Scholars(2020J06014) and Natural Science Foundation of Fujian Province(2018J07005).
通信作者:倪一涛(yitao_ni@fzu.edu.cn)

的鲁棒性提出了一定的要求。

针对以上传统包装器的问题，本文提出了一种基于网页特征相似度计算的包装器自适应技术来提高包装器的鲁棒性。对旧包装器进行分析，获取旧网页中目标数据的特征，同时对新网页进行特征提取，获取新网页的特征集合，根据目标数据特征与新网页特征集合的相似度计算结果，重定位旧包装器的目标数据的所在位置，并得到新网页中与之映射的新目标数据，使旧包装器能够自适应新网页的数据抽取。

2 相关工作

在包装器数据提取和维护的研究上，Chidlovskii[6] 关注包装器的适应性，提出了一些基于语法和逻辑规则的组合和应用。Knoblock 等[7] 开发了一种基于机器学习的系统，用于在从网页提取数据失败的情况下对包装器进行验证和还原。Meng 等[8] 提出了一种模式指导下的包装器维护，利用网页变化后仍然保留的特征来识别新页面中目标数据的位置。但该方法依赖于变化后网页所存在的原目标数据特征来实现目标数据的定位，若原目标数据特征搜索结果为空，则会导致包装器自适应失败。Kowalkiewicz 等[9] 主要关注于利用绝对和相对 XPath 查询的包装器的健壮性[10-12]。Chu[13] 等人利用数据路径匹配技术有效提取数据记录，利用数据路径编码对齐技术有效识别数据项。Liu 等[14] 设计了基于 bootstrapping 方法的领域自适应抽取框架用于 Web 数据提取，从最通用的 XPath 开始，匹配并专门化每个节点，直到协调目标节点来构建包装器，然后根据网页样本学习生成鲁棒性好的包装器作为数据提取模型。但这些方法只是简单地从网页节点的 xpath 特征来考虑，准确度较低。

在一些自适应数据抽取的研究上，Gulhane 等[15] 在 Web 站点中分组结构化相似的页面，标注示例网页用于学习基于 xpath 的抽取规则，通过重用规则从非示例页面中提取目标数据，但如果 Web 页面的结构发生变化，可能需要重新学习规则。Wong 等[16] 提出的 IEKA 方法通过识别出站点不变和可变的性质来获得站点的网页训练集；Yang 等[17] 提出的渐进式 Web 信息抽取，利用关联挖掘算法从训练样本网站中寻找内容关联知识，并利用这种关联知识来识别兴趣信息块，然后通过半指导式学习和无指导式学习，归纳出适用于同一领域的不同网站的抽取规则。但这两种方法须人工辅助，获取相应的训练样本也须耗费较多的时间成本。Deng 等[18] 提出的基于关键词聚类和节点距离的网页信息抽取方法，只要获取到该领域的关键词就能有效地抽取信息，但是该方法仅仅只是针对 HTML 中的〈table〉标签来建立网页结构树，因此不能抽取出非〈table〉标签中的商品信息。Chang[19] 提出了一种自适应包装器生成器，旧包装器通过预存储的结果模式与新 Web 页面进行比较，在新页面中找到所需信息的位置和顺序，并查找所有包含标签来推断新提取规则，但该方法须假设新页面中至少有一条记录存在于预存储的结果模式中。Liu 等[20] 提出了一种基于机器学习技术的最小代价脚本编辑模型，该方法考虑了结构变化下的 3 种编辑操作，即插入节点、删除节点和替换节点标签，通过训练数据生成候选包装器集，再根据成本选取鲁棒性最好的包装器适应未来网页变化的数据提取。Tekale 等人[21] 提出了一种新网站自动学习信息提取知识的系统，通过确定锚标记的 DOM 树路径，再通过该路径识别其他相关属性，从而提取与锚标记相关的信息，但该方法主要针对类似产品展示类型的网站，例如书本或者电子领域，即提取的信息网页具有特定的锚标记。

在网页结构相似度计算的研究上，常用方法是基于树编辑距离[22-23]，然而这种方法不适合用在处理大量网页的情况下，因为其时间复杂度过高。Ferrara 等[24] 提出了一种基于改进树编辑距离匹配技术的相似性度量方法来实现包装器自适应。该方法依赖于利用从旧版本网页中获取的一些结构信息与新版本网页进行比较的可能性，自动重新引入包装器，但该方法只从网页的结构信息来考虑相似度计算，忽略了其他特征对相似度的影响。IBM 实验室的 Joshi 等[25] 提出了基于树的路径匹配来计算网页间的结构相似度的算法。基于树路径匹配算法相对而言时间复杂度较低，但是这个算法的结果显示它的精确度不高。此外，针对树的层次结构，Ferrara 等[26] 提出了一种聚类树匹配算法，该算法通过动态规划产生最大匹配值来计算两棵树之间的相似度，实现起来相比树编辑距离算法更为简单，但这个算法还有一些限制，比如该算法不能匹配节点的排列。

基于以上研究，本文将改进网页间相似度计算方法，从网页的结构信息、视觉信息以及文本信息等多方面特征进行网页相似度的综合计算，提高网页区域匹配的准确性，根据相似度计算结果实现自适应数据抽取。

3 方法概述

如图 1 所示，基于相似度计算的网页包装器自适应主要分为两个模块组成，分别是特征提取模块和包装器重定位与自适应模块。

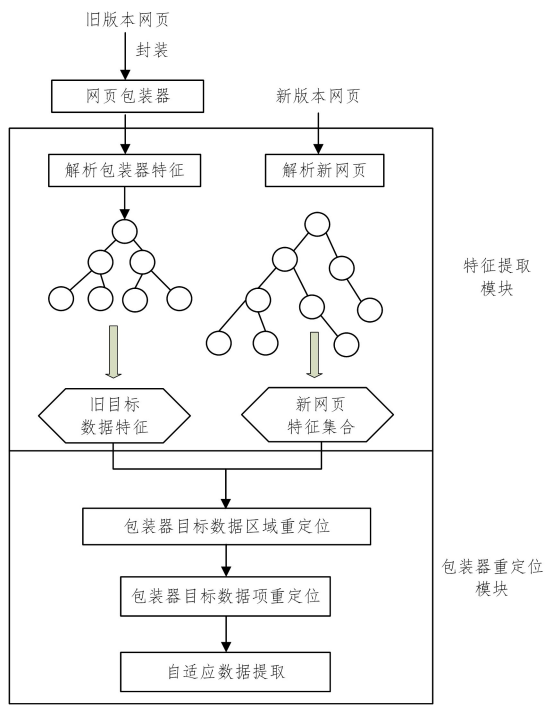

图 1 方法流程

Fig. 1 Method process

其过程可分为 3 个步骤：1）输入旧包装器与新网页地址，解析包装器，得到旧包装器特征子集，通过网页特征提取算法

解析新网页特征，得到新网页特征集合，并以树形结构表示(特征 DOM 树)；2)根据旧包装器特征子集与新网页特征 DOM 树的特征相似度计算结果，得到新网页中与旧包装器映射的目标数据区域(DOM 子树)和目标数据项，从而实现旧包装器的重定位；3)最后根据旧包装器重定位的结果，旧包装器能够在新网页中对新的目标数据进行自适应抽取，实现旧包装器对新网页的自适应。

4 网页特征提取

4.1 特征定义

包装器所要提取的数据往往存在于同一个数据块(同一个 DOM 子树)中，而 DOM 子树中又包含了许多个待提取的数据项。为了能够提高网页包装器自适应结果的准确率，本文将根据基于 DOM 树的多类特征进行相似度综合计算，再根据计算结果完成数据映射匹配，实现包装器的自适应数据提取。其中，无论是数据块还是数据项，特征均分为视觉特征、结构特征以及文本特征。

对于数据块来说，以网页的左上角为坐标原点，建立平面直角坐标系，确定其所在的 x 坐标和 y 坐标；而且每一个数据块以矩形的形式存在，所以数据块的视觉特征由所处坐标以及矩形的大小构成，即可用四元组表示为 $\langle x, y, width, height\rangle$。网页中的数据块是 DOM 树的一棵子树，有其相应的树结构特征，故结构特征由数据块对应的 DOM 子树特征定义，其中包含了根节点路径、树结构和子节点个数，即定义为 $\langle xpath, DOMTree, length\rangle$。数据块内包含了多个节点文本，所以数据块的文本特征则由数据块内所有节点的文本集合 *content* 以及该文本集合的长度 *Len*(*content*)所构成。数据块特征如表 1 所列。

表 1 数据块特征

Table 1 Datablock features

分类	特征	定义
视觉特征	x	距离网页左侧的长度
	y	距离网页顶部的长度
	width	在网页中所占的宽度
	heigh	在网页中所占的高度
结构特征	*xpath*	数据块根节点的路径
	DOMTree	在网页中的 *DOM* 树结构
	length	数据块根节点的子节点个数
文本特征	*content*	数据块内所有文本内容
	Len(*content*)	数据块内所有文本内容长度

对于数据项来说，数据项包含于数据块之内，数据项的位置则以所处数据块的左上角为坐标原点，建立坐标系，确定数据项在数据块内的相对位置 x 和 y；同样，数据项在网页中仍以小矩形存在，故视觉特征由所处数据块内位置和数据项的宽高所定义。数据项是 DOM 树的一个节点，由特有的树节点特征来定义其结构特征，其中包含了节点的块内路径、父节点以及节点的属性集合(如标签名、类名和 id 等)，即结构特征定义为 $\langle xpath, father, Attrs\langle tag, id, class\rangle\rangle$。数据项的文本内容是一个网页标签节点所包含的文本内容，区别于数据块的文本内容集合，故数据项的文本特征是对应网页标签节点所包含的文本内容及文本内容长度。数据项特征如表 2 所列。

表 2 数据项特征

Table 2 Data item features

分类	特征	定义
视觉特征	x	距离数据块左侧的长度
	y	距离数据块顶部的长度
	width	在网页中所占的宽度
	heigh	在网页中所占的高度
结构特征	*xpath*	数据项所在的块内路径
	Attrs	数据项的标签属性集合
	father	数据项节点的父节点
文本特征	*text*	数据项的文本内容
	Len(*text*)	数据项的文本内容长度

4.2 特征提取

利用 JavaScript 可对 DOM 树及其属性进行操作的特性，我们利用 PhamtomJS 无头浏览器加载渲染待提取特征的网页，在内存中形成该网页的 DOM 模型，利用 PhamtomJS 可编程的特性，加载执行节点特征提取算法，获取每一个 DOM 树节点相应的视觉特征、结构特征以及文本特征，并将获取的特征保存为一个新节点，称之为特征 DOM 树节点。节点特征提取算法的伪代码如算法 1 所示。

算法 1 节点特征提取算法(getNodeFeature)

```
输入:DOM 树节点(n)
输出:特征 DOM 树节点(leaf)
1. if node.offsetWidth>0 then
2. /* 获取节点 node 的特征 */
3.    /* 视觉、结构以及文本特征 */
4.    data←getElementTop(node)
5.    data←getElementLeft(node)
6.    data←node.offsetWidth
7.    data←node.offsetHeight
8.    data←node.children/tag/id/class
9.    /* 利用 JavaScript 继续获取 DOM 树节点特征 */
   ……
10. /* 保存特征属性于 leaf 节点 */
11. save data into leaf
12. endif
```

算法首先判断该节点是否为空节点(第 1 行)，然后调用 JavaScript 对 DOM 树节点的操作函数，获取 DOM 树节点相应的特征属性，如节点位置、节点所占的宽高、儿子节点集合、标签名、标签 id 以及类名等(第 4－9 行)，最后将特征属性保存到一个新的树节点，即特征 DOM 树节点 *leaf*(第 11 行)。

算法 1 实现了网页单个节点的特征提取，而对整个网页的特征提取具体是通过对整个网页的 DOM 树采用深度优先遍历，直至完全遍历整个 DOM 树结构，对每个遍历节点调用节点特征提取算法(getNodeFeature)，获取每个节点的视觉、结构和文本特征，并根据网页 DOM 树节点之间的关系，建立与 DOM 树对应的特征 DOM 树，获取整个网页的特征集合。网页特征提取算法的伪代码如算法 2 所示。

算法 2 网页特征提取算法(getHtmlFeature)

```
输入:DOM 树节点(n)
输出:特征 DOM 树节点(root)
1. /* 获取 DOM 树节点的特征 */
2. root←getNodeFeature(n)
3. /* 深度递归访问 DOM 树各节点 */
4. foreach ch in children
5.    child←getHtmlFeature(ch)
```

```
6.  if child!=null then
7.  root.children.push(child)
8. endif
9. endfor
10. return root
```

算法首先调用单个节点的特征提取算法，获取单节点的特征（第2行）；然后采用深度优先遍历DOM树结构，递归调用算法2，获取每个节点的属性特征，从而建立与DOM树相对应的特征DOM树（第4－10行）。

5 包装器重定位与自适应

5.1 包装器定义

基于DOM树的网页包装器主要包含两部分：目标数据（Data）和目标数据的DOM子树（DOMTree）。因此，包装器可形式化定义为一个二元组表示，即 $Wrapper=\langle Data, DOMTree\rangle$。其中，$Data$ 表示包装器所抽取的网页数据项集合，该数据集合包含 n 个数据项，即 $Data=\langle item_1, item_2, \cdots, item_n\rangle$，而每个数据项 $item$ 由数据项名称、数据项值和数据项路径构成，即 $item=\langle name, value, xpath\rangle$。$DOMTree$ 为目标数据对应的网页DOM子树，该子树由 n 个节点构成，即 $DOMTree=\langle node_1, node_2, \cdots, node_n\rangle$，每个 $node$ 节点又包含多个属性，比如节点标签名、节点的父节点、节点的子节点列表、节点的路径、节点的文本内容和节点的特征属性集合，用一个六元组表示为 $node=\langle tag, FatherNode, ChildNode, xpath, text, Attr\rangle$。其中，特征属性集合又包括标签id、标签类名、与页面左边框的距离 x 与网页顶部的距离 y、所占区域的宽度 $width$ 和所占区域的高度 $height$，即 $Attr=\langle id, classname, x, y, width, height\rangle$。

5.2 包装器目标数据区域重定位

网页结构的改变，导致旧包装器面对新网页无法正确抽取数据而失效，须对旧包装器中的原目标数据区域（即 $DOMTree$ 属性）进行重定位，将其重定位到新网页目标数据对应的DOM子树。由于网页DOM子树区域内存在着各类特征，所以本文提出了一种基于网页特征相似度计算的方法，计算新网页中与原目标区域最相似的网页区域，实现目标区域重定位。其中，相似度计算的特征主要包含位置特征、结构特征和文本特征。结构特征能够衡量两个网页区域的DOM树及样式的相似程度；位置特征则表示了网页区域在整个网页DOM树中所处的位置，由于大部分网页的结构变化只会在原区域所在的位置附近移动，因此路径相似度是相似度计算的一个重要指标；文本特征也是网页特征的一个度量因子，但因为网页的文本变化概率较高，且若文本一直保持不变，数据提取也将失去其意义，所以文本相似度在相似度计算中所占的权重较低。

具体实现方法：采用深度优先遍历特征提取后的新网页特征DOM树，根据以遍历节点为根节点的特征DOM子树与原目标区域DOM树的结构相似度、位置相似度和文本相似度的加权计算结果，选择相似度最高且大于阈值（0.65）的特征DOM树作为旧包装器的映射区域 $newDOMTree$，即实现旧包装器的目标区域重定位。其中，相似度阈值是通过多次实验分析确定，当相似度阈值设置为0.65时，包装器自适应的效果较好。

由此，网页区域 i 和网页区域 tar 的相似度计算公式为：

$$similarity(i,tar)=sim(P_i,P_{tar})\times 0.2+sim(T_i,T_{tar})\times 0.7+sim(Text_i,Text_{tar})\times 0.1 \quad (1)$$

其中，$sim(P_i,P_{tar})$ 表示两个网页区域的位置相似度，$sim(T_i,T_{tar})$ 表示两个网页区域的结构相似度，$sim(Text_i,Text_{tar})$ 表示两个网页区域的文本相似度。

5.2.1 区域位置相似度计算

区域的位置特征由该区域DOM子树在网页DOM树中的位置来表示。利用网页的xpath路径能够唯一定位到网页DOM树中一个特定节点的特性，本文将区域根节点相距网页根节点的xpath路径作为该数据区域在整个网页中所处的位置。

传统的xpath路径匹配算法只考虑标签名序列的匹配，忽略了该标签的具体位置信息（如在兄弟节点的次序），准确度较低。例如，路径 p_1：/html/body/div/div 虽然能够准确定位图2网页变化前的目标区域（下划线div），但当网页结构变化为如图3所示时，则传统xpath路径无法准确定位新网页的目标区域。

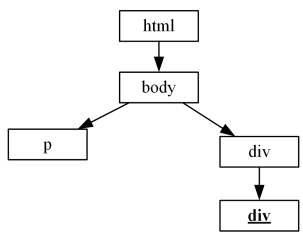

图2 网页变化前的DOM树

Fig. 2 DOM tree before change

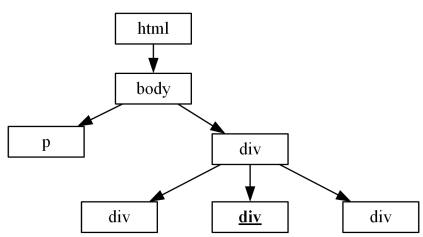

图3 网页变化后的DOM树

Fig. 3 DOM tree after change

因此，每一个xpath路径不仅要包含节点的标签名顺序，还应保存每一个节点在其兄弟节点中的次序，例如路径 p_2：/html[0]/body[0]/div[0]/div[1]中包含网页根节点到区域DOM子树根节点的节点标签顺序，其后数字表示该节点在其兄弟节点中的次序，且 p_2 能够准确定位图3变化后的网页目标区域。本文从标签名顺序以及标签在其兄弟节点中的次序两方面进行综合考虑，提出了改进后的路径匹配算法来计算区域位置相似度。

给定两个xpath路径：

$p_i: tag_{i,1}[x_1]/tag_{i,2}[x_2]/\cdots/tag_{i,n}[x_n]$

$p_j: tag_{j,1}[y_1]/tag_{j,2}[y_2]/\cdots/tag_{j,n}[y_n]$

则路径 p_i 和路径 p_j 的相似度计算公式为：

$$sim(P_i,P_j)=\frac{L}{\min(len(P_i),len(P_j))}\times 0.6+\frac{P}{L} \quad (2)$$

其中，$len(p_i)$ 为计算路径 p_i 的节点个数，L 是以 tag_1 为起始点的最长标签公共序列长度，P 是以 tag_1 为起始点的最长标签公共序列且满足兄弟节点次序相同的节点个数。

5.2.2 区域结构相似度计算

网页中每一个区域都对应网页 DOM 树中的一棵子树，并且每个网页区域都有其不同的视觉特征，如区域的宽度和高度。因此，区域结构相似度由两个区域的 DOM 子树结构相似度 $st(T_i, T_{tar})$ 以及两个区域的可视化结构相似度 $sp(T_i, T_{tar})$ 共同决定。其中，$st(T_i, T_{tar})$ 衡量两个区域的标签结构、DOM 子树及区域布局的相似性，$sp(T_i, T_{tar})$ 衡量两个区域的样式特征相似性。由此，两个区域 T_i 和 T_{tar} 的结构相似度计算公式为：

$$sim(T_i, T_{tar}) = st(T_i, T_{tar}) \times 0.5 + sp(T_i, T_{tar}) \times 0.5 \tag{3}$$

两个区域的 DOM 子树的结构相似度 $st(T_i, T_{tar})$ 是根据两个区域的 DOM 子树特征进行计算的，包含了根节点标签类型一致性(st_1)、子树节点个数比(st_2)、子树高度比(st_3)以及树结构相似度(st_4)。其中，当两个区域子树根节点标签类型一致时 $st_1=1$，否则 $st_1=0$；子树节点个数比 st_2(子树高度比 st_3)都是由两棵子树中最小节点数(最小高度)除以最大节点数(最大高度)得到的，如式(4)所示。

$$st_{2,3}(T_i, T_{tar}) = \frac{\min(T_i, T_{tar})}{\max(T_i, T_{tar})} \tag{4}$$

其中，T_i 和 T_{tar} 是区域 i 和区域 tar 的 DOM 子树，$\min(T_i, T_{tar})$ 计算两个区域子树的最少节点数或最小高度，$\max(T_i, T_{tar})$ 计算两个区域子树的最大节点数或最大高度，$st_{2,3}(T_i, T_{tar})$ 衡量了两个子树规模的相似度。

树结构相似度 $st_4(T_i, T_{tar})$ 采用聚类树匹配算法。算法的具体思想为：若两棵子树节点规模相差超过 10 倍则返回 0；否则采用动态规划算法得出两棵子树具有相同标签节点的最大数目 $maxNode$，最后计算相同标签数占两棵子树中最大节点数的比值，该比即值为两棵子树的树结构相似度。算法伪代码流程如算法 3 所示。

算法 3 聚类树匹配算法

输入：两棵 DOM 子树 T_i 和 T_j，节点个数 m 和 n

输出：两棵子树的结构相似度 $st_4(T_i, T_{tar})$

```
1. for all p suchthat 1≤p≤m do
2.    for all q suchthat 1≤q≤n do
3.       M[i][j]←max(M[p][q−1],M[p−1][q],M[p−1][q−1]+
         W[p][q]) Where W[p][q]=match(T_i(p−1),T_tar(q−1))
4. endfor
5. endfor
6. if m>0 and n>0 then
7. return st_4=M[m][n]/max(t(T_i),t(T_tar))
```

设置矩阵 M，矩阵 $M[i,j]$ 表示第一棵子树的前 i 个节点与第二棵子树的前 j 个节点中最大的相同标签节点数，采用动态规划遍历两棵子树，根据动态规划关系式 $M[p][q]=\max(M[p][q-1], M[p-1][q], M[p-1][q-1]+W[p][q])$ 计算出 $M[m,n]$。$\max(t(T_i), t(T_{tar}))$ 用于计算两个子树中最大的节点数。

由此，两个区域的 DOM 子树结构相似度 $st(T_i, T_{tar})$ 计算公式为：

$$st(T_i, T_{tar}) = st_1 \times 0.1 + st_2 \times 0.2 + st_3 \times 0.2 + st_4 \times 0.5 \tag{5}$$

两个区域的可视化结构相似度 $sp(T_i, T_{tar})$ 计算的是两个区域视觉特征的相似度，包含了左边距相似比(R_1)、上边距相似比(R_2)、区域宽度相似比(R_3)和区域高度相似比(R_4)。这四者的相似比计算，都是由两个区域对应特征的最小值除以两个区域对应特征的最大值，如式(6)所示。

$$R_{1,2,3,4}(D_i, D_{tar}) = \frac{\min(D_i, D_{tar})}{\max(D_i, D_{tar})} \tag{6}$$

其中，D_i 和 D_{tar} 表示网页区域 i 和网页区域 tar。$\min(D_i, D_{tar})$ 计算网页区域 i 与网页区域 tar 在某个视觉特征下的最小值，$\max(D_i, D_{tar})$ 计算网页区域 i 与网页区域 tar 在某个视觉特征下的最大值。例如，当计算左边距相似比(R_1)时，$\min(D_i, D_{tar})$ 表示两个区域中与网页左边栏最小的距离值，而 $\max(D_i, D_{tar})$ 表示两个区域中与网页左边栏最大的距离值，相似比即为两者的比值。

由此，可视化结构相似度 $sp(T_i, T_{tar})$ 的计算公式为：

$$sp(T_i, T_{tar}) = R_1 \times 0.25 + R_2 \times 0.25 + R_3 \times 0.25 + R_4 \times 0.25 \tag{7}$$

5.2.3 区域文本相似度计算

区域文本是区域子树上所有节点文本的集合，文本相似度由文本语义相似度(T_1)和文本长度相似度(T_2)加权计算衡量，则文本相似度计算公式为：

$$sim(Text_i, Text_{tar}) = T_1 \times 0.4 + T_2 \times 0.6 \tag{8}$$

其中，语义相似度计算首先通过开源分词工具(IKAnalyzer)进行文本分词。根据过词向量间的余弦相似度，来计算每对词之间的语义相似度。区域 i 与目标区域 tar 的语义相似度的计算是通过遍历区域 i 中的每个词 $text_{i,j}$，在区域 tar 中寻找与之相似度最高的词，计算两词之间的余弦相似度 $sim(text_{i,j}, con_{tar})$ 作为该词的相似度，最后取区域 i 所有词相似度的平均值作为该区域与目标区域 tar 的语义相似度 T_1。计算公式如下：

$$T_1 = \sum_{j=0}^{k} sim(text_{i,j}, con_{tar}) \times \frac{1}{k} \tag{9}$$

$$\mathrm{sim}(text_{i,j}, con_{tar}) = \max(sim(text_{i,j}, con_{tar,k})) \tag{10}$$

其中，$k=1,2,\cdots,n$；$text_{i,j}$ 是一个节点包含的文本内容，表示区域 i 中的第 j 个文本；con_{tar} 表示区域 tar 的文本集合；$sim(text_{i,j}, con_{tar})$ 计算区域 i 中的第 j 个词与区域 tar 内各个词的相似度，返回其中最大的相似度。

文本长度相似度的计算主要依赖于两个区域文本集合的长度比，公式如下：

$$T_2 = \frac{\min(con_i.len, con_{tar}.len)}{\max(con_i.len, con_{tar}.len)} \tag{11}$$

其中，$con_i.len$ 是区域 i 的文本长度，$con_{tar}.len$ 是区域 tar 的文本长度。

5.3 包装器目标数据项重定位

由目标区域重定位确定了旧包装器在新网页中的映射区域($newDOMTree$)，即确定了新网页中待提取数据的网页 DOM 子树。为了确定新网页中的待提取数据项与旧包装器数据项之间的映射关系，通过旧包装器数据集合 $Data$ 中的每一个数据项 $item$ 与新网页映射区域的各个数据项进行相似度计算，得到新网页目标数据区域中与旧包装器匹配的数据集合 $newData$，实现目标数据项重定位。

数据项间的相似度仍然由结构相似度、路径相似度和文本相似度三者加权计算衡量(即式(1))。其中路径相似度、文本相似度的计算与数据区域相似度计算相同(即式(2)和式(8))，区别在于数据项路径是相对于区域根节点，即 $path=$

$xpath_i - xpath_{root}$，$xpath_{root}$为该区域的根节点路径，$xpath_i$为该节点相对于网页根节点的 xpath 路径；数据项文本相似度是单节点文本之间的相似度，即文本集合中只有一个元素。

数据项是 DOM 树上的一个节点，数据项结构特征包含了该数据项所处的 DOM 树节点的各类特征，并由这些特征来计算两个数据项之间的结构相似度。其中包含了标签、标签 id、类名和父节点的一致性计算(s_1)，区域内的相对位置 X，Y 坐标相似性(s_2)，以及数据项节点区域大小相似性(s_3)。其中，特征一致性(s_1)的计算是当两个数据项在某一特征的取值一致时，则表示特征一致，即$s_1=1$；否则$s_1=0$。例如，当两个数据项的标签名一致时，$s_1(tag)=1$。而坐标相似性(s_2)以及区域大小相似性(s_3)的计算则与两个数据项在该特征的最大值和最小值有关，计算公式如式(12)和式(13)所示。

$$s_2=\frac{\min(x_i,x_j)}{\max(x_i,x_j)}\times 0.5+\frac{\min(y_i,y_j)}{\max(y_i,y_j)}\times 0.5 \tag{12}$$

$$s_3=\frac{\min(w_i,w_j)}{\max(w_i,w_j)}\times 0.5+\frac{\min(h_i,h_j)}{\max(h_i,h_j)}\times 0.5 \tag{13}$$

其中，x_i表示数据项 i 的 x 坐标，y_i表示数据项 i 的 y 坐标；w_i表示数据项 i 占据的宽度，h_i表示数据项 i 占据的高度；min()计算两个数据项在某一特征下的最小值，max()计算两个数据项在某一特征下的最大值。

将数据项各个结构特征相似度计算结果加权求和，即得数据项结构相似度计算公式：

$$sim(data_i,data_{tar})=(s_1(tag)+s_1(id)+s_1(class)+s_1(father))\times 0.1+(s_2+s_3)\times 0.3 \tag{14}$$

最后，当完成包装器的目标区域和目标数据重定位后，得到了新网页中新的待抽取目标数据信息，包括目标数据的 DOM 子树 *newDOMTree* 和目标数据项集合 *newData*，使得包装器能够根据新网页目标数据信息进行自适应数据抽取。

6 实验

6.1 实验准备

为了验证本文所提出的基于相似度计算的网页包装器自适应的有效性，本文实验从网站历史博物馆[1)]选取了 5 种类型网站，其中包括购物类、新闻类、资讯类、论坛类和服务类，每一种类型选取了 50 对网页，一共选取了 250 对网页进行了包装器自适应实验，其中每对网页包含其新旧版本 2 个网页则共有 500 个网页，如表 3 所列。其中，5 种类型的网站主要涵盖了两大网页设计类型，包括功能型网页设计(例如服务类、购物类)和信息型网页设计(例如新闻类、资讯类等)。

表 3 实验样本

Table 3 Experimental sample

网站类型	模板数量	网页数量(包含新旧两版网页)
购物类	5	50(100)
新闻类	5	50(100)
资讯类	5	50(100)
论坛类	5	50(100)
服务类	5	50(100)

本文实验分为两部分：第一个部分是根据旧版本网页生成的网页包装器对新网页进行自适应数据提取实验，一共实验了 250 对，每一对中包含旧网页包装器和新版本网页，并根据自适应的成功率 $S(N)$以及数据提取的精确度 $P(N)$、召回率 $R(N)$和 F 值来评价包装器自适应的有效性；第二部分是将本文方法与现有的两个包装器自适应技术进行实验对比，通过设置 4 个不同的数据源，进行数据提取实验，根据数据提取的精确度 $P(N)$、召回率 $R(N)$以及 F 值来评价本文方法与现有方法之间的优点。其中，实验中使用的评估计算方法如式(15)－式(18)所示。

成功率：

$$S(N)=\frac{T}{Total} \tag{15}$$

其中，T 表示可以成功自适应的网页数，$Total$ 表示总的实验样本数。

精确度：

$$P(N)=\frac{TP}{TP+FP} \tag{16}$$

其中，TP(True Positives)表示正类被判定为正类，即正确找到目标信息的数量；FP(False Positives)表示负类被判定为正类，即找到的信息并非是目标信息的数量。

召回率：

$$R(N)=\frac{TP}{TP+FN} \tag{17}$$

其中，FN(False Negatives)表示正类判定为负类，即没有被正确找到的目标信息数量；$TP+FN$ 就是目标信息的总数。

F 值：

$$F=\frac{2\times P\times R}{P+R} \tag{18}$$

6.2 包装器自适应数据抽取

对 5 种类型网站的 250 个旧版本网页的包装器，根据相应的新版本网页进行自适应数据抽取，统计可自适应的网页数量，计算相应的成功率 $S(N)$，并对数据提取的精确度、召回率及 F 值进行分类讨论，实验结果如表 4 和表 5 所列。

表 4 包装器自适应实验结果

Table 4 Adaptive experiment results

网站类型	网页数量	可自适应网页数	S(N)/%
购物类	50	39	78.0
新闻类	50	46	92.0
资讯类	50	40	80.0
博客类	50	47	94.0
服务类	50	45	90.0

表 5 数据提取实验结果

Table 5 Experimental results of data extraction

网站类型	网页数量	P(N)/%	R(N)/%	F 值/%
购物类	50	71.8	74.1	72.9
新闻类	50	89.9	91.1	90.5
资讯类	50	73.6	77.1	75.3
博客类	50	90.4	92.3	91.3
服务类	50	85.3	87.5	86.4

实验结果表明，购物类和资讯类网页的包装器自适应成功率较低，自适应效果较差，但总体平均自适应成功率仍然在 83%左右。从表 5 可以看出，资讯类和购物类网站数据提取的准确率也较低。通过观察其新旧版本的网页，可以得出原因在于这两类网页的变化幅度较大，资讯类网站和购物类网

1) https://web.archive.org/

站的网页结构通常会发生较大的改变，同时网页中的干扰信息较多，从而自适应效果较差，但数据提取的平均精确度和平均召回值分别能达到 82.2%和 84.36%，说明本文提出的方法仍具有有效性，但更适用于新闻类、博客类及服务类网站的数据提取。

6.3 网页自适应数据提取对比实验

将本文提出的方法与现有方法 SG-WRAP 以及 Emilio 等提出的通过树编辑距离匹配自动调整包装器的包装器维护方法 TEDM(Tree Edit Distance Matching)进行实验对比。实验采用了 Allbooks4-less，Hotels，Amazon 和 Barnesand-noble 4 个数据源进行实验，并根据数据提取的精确度 $P(N)$、召回率 $R(N)$以及 F 值来对比评价 3 个方法在包装器自适应技术上的有效性，实验结果如表 6－表 8 所列。

表 6 SG-WRAP 方法的实验结果

Table 6 Experimental results of SG-WRAP method

(单位：%)

数据源	$P(N)$	$R(N)$	F 值
Allbooks4-less	51.3	75.0	60.9
Hotels	41.9	50.0	45.6
Amazon	90.7	83.1	86.7
Barnesand-noble	100	78.7	88.1
平均	71.0	71.7	70.3

表 7 TEDM 方法实验结果

Table 7 Experimental results of TEDM method

(单位：%)

数据源	$P(N)$	$R(N)$	F 值
Allbooks4- less	76.9	98.5	86.4
Hotels	72.4	62.5	67.1
Amazon	92.1	92.1	92.1
Barnesand- noble	93.8	95.3	94.5
平均	83.8	87.1	85.0

表 8 本文方法的实验结果

Table 8 Experimental results of proposed method

(单位：%)

数据源	$P(N)$	$R(N)$	F 值
Allbooks4- less	92.6	99.3	95.8
Hotels	79.5	85.3	82.3
Amazon	94.7	97.3	95.9
Barnesand- noble	96.9	98.2	97.5
平均	90.9	95.0	92.9

SG-WRAP 利用有效的数据特征对网页包装器进行维护更新，其数据特征包含语法特征、超链接和提取数据项的注释，用这些特征寻找所需数据项在更新后的网页中所处的位置。实验结果表明，该方法在一些网站的数据提取精确率和召回率达到了理想值，但其存在一定的局限性，不适用于大部分网站，其在“Hotels”网站的数据提取精确率及召回率仅为 41.9%和 50%，并且该网站的大多数数据结构及网页结构都没有改变，只是其他的数据特征(注释和超链接)没有保留。所以，SG-WRAP 方法在网页结构变化下，自适应数据提取具有较大的局限性。

TEDM 依赖于利用从旧版本网页中获取的一些 DOM 树的结构信息与新版本网页对应的结构信息进行比较，从而自动重新引入包装器。实验结果表明，TEDM 方法的平均精确度相对于 SG-WRAP 有了一些提升，但该方法只从新旧版本网页的结构信息来考虑相似度计算，忽略了其他特征对相似度影响，因此 TEDM 方法在平均精确度、平均召回率及 F 值上的表现不及本文所提出的方法。

结束语 本文提出了一种基于网页特征相似度计算的包装器自适应技术，致力于解决传统包装器因为网页结构的变化致使原有包装器失效的问题。该方法首先提取变化后的网页特征以及旧包装器所抽取的目标数据特征，遍历新网页节点，计算节点与原目标数据在结构、位置和文本 3 方面的特征相似度，得到与原目标数据匹配的数据区域，实现旧包装器的重定位，最后根据重定位结果，对新网页进行自适应数据抽取。实验验证，本文方法具有良好的可行性。

但本文的研究还存在一些不足之处，今后将进一步完善相似度计算，将网页中更多的相关特征纳入相似度考虑范畴，提高相似度的准确性，并进行更多的实验来进一步评估方法的有效性。

参 考 文 献

[1] CNNIC's 45th Statistical Report on the Development of China's Internet[EB/OL]. http://www.cac.gov.cn/2020-04/27/c_1589535470378587.html.

[2] CUI C, GONG J. Overview of Web Information Extraction Research[J]. Computer Knowledge and Technology: Academic Exchange, 2011, 7(4): 2279-2280.

[3] CAFARELLA M J, HALEVY A Y, WANG D Z, et al. WebTables: Exploring the power of tables on the web[J]. Proceedings of the VLDB Endowment, 2008, 1(1): 538-549.

[4] ZHANG J. Research and Implementation of Web Information Automatic Extraction Technology[D]. Wuhan: Wuhan University of Technology, 2009.

[5] EMILIO F, ROBERT B. Automatic Wrapper Adaptation by Tree Edit Distance Matching[C]// Proceedings of the 2nd International CIMA Workshop. Springer, 2011: 41-54.

[6] CHIDLOVSKII B. Automatic Repairing of Web Wrappers[C]// Proceeding of the Third International Workshop. ACM, 2001: 24-30.

[7] KNOBLOCK C A, LERMAN K, MINTON S N. Wrapper Maintenance: A Machine Learning Approach[J]. Computer Science, 2011, 18(1): 2003.

[8] MENG X, HU D, LI C. Schema-guided wrapper maintenance for web-data extraction[C]// Fifth ACM CIKM International Workshop on Web Information and Data Management. ACM, 2003: 1-8.

[9] KOWALKIEWICZ M, KACZMAREK T, ABRAMOWICZ W. myPortal: Robust Extraction and Aggregation of Web Content[C]// Proceedings of the 32nd International Conference on Very Large Data Bases. DBLP, 2006: 1219-1222.

[10] DALVI N N, BOHANNON P, SHA F. Robust web extraction: an approach based on a probabilistic tree-edit model[C]// ACM Sigmod International Conference on Management of Data. ACM, 2009: 335-348.

[11] LEOTTA M, STOCCO A, RICCA F, et al. Reducing Web Test Cases Aging by Means of Robust XPath Locators[C]// IEEE International Symposium on Software Reliability Engineering Workshops. IEEE, 2014: 449-454.

(下转第 257 页)

[12] KONG C Y,YU J. Sentiment analysis of real estate agency reviews combining semantic rules and sentiment dictionary[J]. Information Technology And Informatization,2020(4):20-24.
[13] LI X,XIE H,LI L J. Study on the calculation of sentence semantic similarity based on Word2vec[J]. Computer Science,2017,44(9):256-260.
[14] SHARMA A K,CHAURASIA S,SRIVASTAVA D K. Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec[J]. Procedia Computer Ence,2020,167:1139-1147.
[15] LI W,QI F,TANG M,et al. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification[J]. Neurocomputing,2020,387:63-77.
[16] LÜ W,LI Z,CHU J. Adaptive Ensemble Undersampling-Boost: A Novel Learning Framework for Imbalanced Data[J]. Journal of Systems & Software,2017,132(10):272-282.

WANG You-wei, born in 1987, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include machine learning and data mining.

ZHU Chen, born in 1992, postgraduate. Her main research interests include data mining and natural language processing.

(上接第 224 页)

[12] LIU D,WANG X,LI H,et al. Robust Web Extraction Based on Minimum Cost Script Edit Model[J]. Procedia Engineering,2012,29(1):1119-1125.
[13] CHU Y C,HSU C C,LEE C J,et al. Automatic data extraction of websites using data path matching and alignment[C]// Fifth International Conference on Digital Information Processing & Communications. IEEE,2015.
[14] LIU D L,LIU X,MA L,et al. Domain adaptation of web data extraction based on bootstrapping method[C]// International Conference on Electronics. 2017.
[15] GULHANE P,MADAAN A,MEHTA R,et al. Web-scale information extraction with vertex[C]// 2011 IEEE 27th International Conference on Data Engineering. IEEE,2011:1209-1220.
[16] WONG T L,LAM W. Adapting Web information extraction knowledge via mining site-invariant and site-dependent features [J]. ACM Transactions on Internet Technology,2007,7(1):6.
[17] YANG P,ZHENG Q L,PENG H,et al. A stepwise learning approach to automatic discovery of interest data blocks[C]// Proceedings of 2004 International Conference on Machine Learning and Cybernetics. IEEE,2004:1441-1446.
[18] DENG J S,ZHENG Q L,PENG H. Web page information extraction based on keyword clustering and node distance[J]. Computer Science,2007(4):217-220.
[19] CHANG Y S. Adaptable wrapper generation for web page format change[C]// Proc. 5th Int. Conf. on Applied Computer Science. World Scientific and Engineering Academy and Society, Stevens Point,Wisconsin,USA,2006:147-152.
[20] LIU D,MA L,LIU X. Research on Adaptive Wrapper in Deep Web Data Extraction[C]// International Conference on Internet of Vehicles. Cham:Springer,2015:409-423.
[21] TEKALE A A,NANDGAONKAR S S. Automatic wrapper adaptation system[J]. International Journal of Scientific & Engineering Research,2013,4(3):7.
[22] REIS D C,GOLGHER P B,SILVA A S,et al. Automatic web news extraction using tree edit distance[C]// Proceedings of the 13th International Conference on World Wide Web (WWW 2004). ACM,2004:502-511.
[23] KIM Y,PARK J,KIM T,et al. Web Information Extraction by HTML Tree Edit Distance Matching[C]// Proceedings of the 5th International Conference on Convergence Information Technology. ACM,2007:2455-2460.
[24] FERRARA E,BAUMGARTNER R. Automatic wrapper adaptation by tree edit distance matching[M]// Combinations of IntelligentMethodsand Applications. Berlin:Springer,2011:41-54.
[25] JOSHI S,AGRAWAL N,KRISHNAPURAM R,et al. A bag of paths mode! for measuring structural similarity in Web documents[C]// Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2003:577-582.
[26] FERRARA E,DE MEO P,FIUMARA G,et al. Web data extraction,applications and techniques: A survey[J]. Knowledge-Based Systems,2014,70:301-323.

CHEN Ying-ren, born in 1997, postgraduate. His main research interests include software adaptation and knowledge mapping.

NI Yi-tao, born in 1969, Ph.D, is a member of China Computer Federation. His main research interests include software engineering, system security and so on.