基于网页聚类的 Web 信息自动抽取*

邱韬奋,杨天奇,曾洪波
(暨南大学 信息科学技术学院计算机系,广东 广州 510632)

摘 要:针对现今较流行的动态 Web 网页数量巨大、数据价值高,并且网页结构高度模板化的特点,设计了一个基于网页聚类的 Web 信息自动抽取系统。在 DOM 抽取技术基础上利用网页聚类寻找高相似簇,并引入列相似度和全局自相似度计算方法,提高了聚类结果的准确性。抽取模板中应用了可选节点对模板的修正和调整,以提高内容节点的正确标识。实验结果表明,该方法能够自动寻找并抽取网页主要信息,达到了较高的准确率和查全率。

关键词:Web 信息抽取;网页聚类;包装器生成
中图分类号:TP391 **文献标识码**:B **文章编号**:1674-7720(2011)04-0071-04

Automatic Web information extraction based on page clustering

Qiu Taofen, Yang Tianqi, Zeng Hongbo
(Dept. of Computer, College of Information Science and Technology, Jinan University, Guangzhou 510632, China)

Abstract: Dynamic Web page has a large amount of pages, high-value data and high-modularity structure. According to these feature, this paper developed an automatic Web information extraction system based on page clustering. On the basis of DOM extraction technique, it used page clustering to find the high similarity clusters, and improved the accuracy of clustering results by using the column similarity measure and global auto-similarity measure. Extraction template applied the optional nodes to modify and adjust the template in order to improve the identification of the content nodes. Experimental result shows this method automatically locates and extracts the main information of pages and achieves high precision and recall.

Key words: Web information extraction; page clustering; wrapper generation

随着 Internet 技术的迅速发展,Web 已经成为当今最庞大的信息库。然而 Web 页面中通常含有很多用户并不关心的信息(如广告链接、导航栏和版权信息等),如何从 Web 页面中抽取出有用的信息已经成为当前信息领域的研究热点之一。

Web 信息抽取是一种从 Web 文档中抽取出有用信息的技术,通常用于 Web 信息抽取的软件又称作包装器(Wrapper)。自 1994 年起,包装器生成技术经历了从手工编写包装器脚本,到利用机器学习的半自动化生成,再到自动化生成的三个阶段。目前,自动化已经成为 Web 信息抽取技术的一个重要特征,比较有代表性的抽取工具有 RoadRunner、IEPAD、Dela 和 MDR-2 等[1]。

1 总体流程

本文根据数据提供网站动态网页的特点,在基于 DOM 的抽取技术上,根据树的相似度比较算法对网页进行聚类分析,从而分类出网页结构相似度较高的网页簇,并考虑非模板标签和模板文本改进包装器生成算法,基于这些算法实现一个高精度的 Web 信息自动抽取系统,并通过大量的测试网页集对这些算法进行实验和评估。整个抽取流程如图 1 所示。

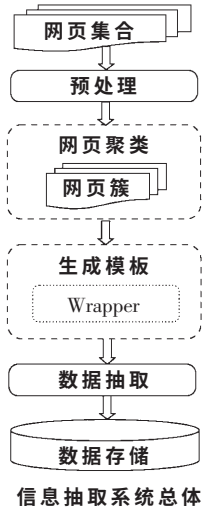

图 1 信息抽取系统总体流程

2 算法实现

2.1 页面预处理

对于抓取的网页,并不能

* 基金项目:广东省科技计划项目(2009B070300052)

直接转化成一个 DOM 树，因为 HTML 网页的格式通常不是规范的 XML 格式，因此需要将其先转化成 XHTML 格式。另外，Web 中很多的网页都会存在标签上的错误，由于 HTML 的不规范性导致代码中存在的标签不配对也不影响页面的执行，并且很多标签是多余的。对于这些问题，本文采用 HTML Tidy[2]来解决。Tidy 是一个开源的 HTML 网页净化工具，它可以将 HTML 转化成 XHTML，并能清除网页中的明显错误。因为页面预处理不是本文的研究重点，所以不对此问题展开阐述。

2.2 树编辑距离

基于 DOM 模型的 Web 信息抽取技术的基础算法就是比较两棵 HTML 标签树的相似性。比较两棵树相似性的方法之一就是计算它们的编辑距离，要计算两棵树之间的树编辑距离[3]，通常的做法是找到两棵树之间权值最小的一个映射(mapping)，映射的定义如下：

定义 1：假设 X 是一棵树，$X[i]$是树 X 中第 i 个字节点，则树 T_1 和 T_2 之间的映射 M 满足以下条件的有序数对(i,j)的集合。

对于任何 M 中的所有有序数对(i_1,j_1)、(i_2,j_2)：

(1)$i_1=i_2$ 的条件是当且仅当 $j_1=j_2$；

(2)$T_1[i_1]$是 $T_1[i_2]$的左邻节点的条件是当且仅当 $T_2[j_1]$是 $T_2[j_2]$的左邻节点；

(3)$T_1[i_1]$是 $T_1[i_2]$的父节点当且仅当 $T_2[j_1]$是 $T_2[j_2]$的父节点。

有了映射，就可以计算两棵树之间的树编辑距离。设两棵树 T_1 和 T_2 之间的映射为 M。在 M 包含的数据对(i,j)中 i、j 分别表示 T_1 和 T_2 的标签。令 S 表示 i 和 j 不相同的数据对数量，即需要替换的标签；D 表示 T_1 中没出现在 M 中的节点，即需要删除的标签；I 表示 T_2 中没出现在 M 中的节点，即需要插入的标签。则树编辑距离 $ed(T_1,T_2)$可以由式(1)得出：

$$ed(T_1,T_2)=Sp+Dq+Ir \quad (1)$$

其中，p、q、r 分别表示替换、删除和插入的权值。为了使得出的值介于 0 与 1 之间，利用式(2)规范化计算结果便于计算相似矩阵，式中的 $len(T_1)$和 $len(T_2)$分别表示 T_1 和 T_2 的节点数：

$$es(p_1,p_2)=1-\frac{ed(T_1,T_2)}{len(T_1)+len(T_2)} \quad (2)$$

2.3 使用代表点的聚类法

对于网页集合的聚类，传统的层次聚类方法能实现比较好的结果。层次聚类过程由不同层次的分割聚类组成，层次之间的分割具有嵌套的关系，整个过程为一个树状结构。自底向上的层次算法称为凝聚层次算法，本文使用的凝聚层次算法是使用代表点的聚类法。

使用代表点的聚类法(Clustering Using Representatives)，首先把每个单独的数据对象作为一个簇，每一步距离最近的簇对首先被合并，直到簇的个数为 K，算法结束。CURE 的显著特点是：(1)可以识别任意形状的簇；(2)有效处理异常数据[4]。

2.4 网页聚类算法

在聚类实验中网页的数目为 500~1000，在这个复杂度上，可以采用类 CURE 算法。在本文的网页聚类实验中，距离定义为两个网页的树编辑距离。

定义网页集合 $P=\{P_0,P_1,\cdots,P_n\}$，根据网页间的 HTML 标签树的树编辑距离计算相似矩阵。将 P_i 和 P_j 利用 $es(p_i,p_j)$计算出树编辑距离，在矩阵中表示为 m_{ij}，计算结果为一个 $N\times N$ 矩阵。定义 P_i 和 P_j 的列相似度为 $cs(p_i,p_j)$，通过实验可以发现引入平均绝对误差概念的列相似度比用单纯的树编辑距离 $es(p_i,p_j)$在聚类过程的计算中具有更好的健壮性。列相似度 $cs(p_i,p_j)$由式(3)得出：

$$cs(p_i,p_j)=1-\sum_{k=1..n}\frac{|m_{ik}-m_{jk}|}{n} \quad (3)$$

紧接着利用代表点的聚类算法对网页进行聚类计算。网页的聚类分析中，首先认为每个网页就是单独的一个簇，然后根据簇间的相似性不断地合并簇对，直到合并为理想的簇的个数时算法结束。这里引用了自相似度的概念以获得更好的聚类结果[5]，其中集合 Φ 的自相似性 $s(\Phi)$由式(4)得出：

$$s(\Phi)=\frac{2}{|\Phi|(|\Phi|-1)}\sum_{p_i,p_j\in\Phi}cs(p_i,p_j) \quad (4)$$

网页聚类中产生的代表簇必须满足两个阈值。首先簇的全局自相似性必须满足阈值 Ω_g，其次簇中两两网页间的列相似度必须满足阈值 Ω_e，这个阈值的设定是为了避免出现新簇，虽有较高的全局自相似性，但簇内仍然包含了一些不相似对象的情况。在本实验中，将 Ω_g 和 Ω_e 值分别设置为 0.9 和 0.8，整个过程算法的伪代码如下：

```
ClusterPage(pageSet, Ωg, Ωe)
    let mij be the distance of Pi and Pj in the pageSet
    Initialize each page to a group and put it into the set of groups G
    while (G>1) do
        choose A, B ∈ G, a pair of groups which maximize
                        the auto-similarity measure s(A∪B)
        if s(A∪B)>Ωg && ∀i,j ∈ A∪B, cs(i,j)>Ωe then
            remove A and B from G
            let Φ=A∪B
            insert Φ into G
        else break
    end while
    return G
```

2.5 抽取模板生成

对于网页聚类后的每一个网页簇，都会生成一个对应的抽取模板，所有抽取模板组成了抽取系统的包装器。网页模板生成建立在两个网页模板生成的基础上。

2.5.1 两个网页的模板

两个网页模板的生成算法的基础也是 DOM 树的相似性算法，在计算距离的同时，生成一个节点映射集合，获得树节点 T_1 和 T_2 之间距离最小的子树匹配情况，把这些匹配情况作为一个列表返回。当 T_1 和 T_2 不匹配时，返回的列表为空；当 T_1 和 T_2 至少有一个没有子节点时，返回的列表只包含 T_1 和 T_2 的匹配。

返回的两个网页的节点映射集合中的节点就是模板中的必需节点，而两个网页不在映射集合中的点可以认为是可选节点，也可以认为是内容节点。如果是可选节点，就要把这些节点插入到模板中，可以把 T_1 认为是最终模板，然后把 T_2 的可选节点插入到 T_1 中。插入的算法是：对于任一 T_2 在映射中的节点 P，获得它在 T_1 中的对应节点 Q，遍历 P 的所有子节点 C，如果节点 C 在 T_1 中存在映射节点 D，则记录 D 节点在 Q 节点的子节点列表中的位置；如果节点 C 在 T_1 中不存在映射，则把节点 C 插入列表中最近一次记录的位置后面。

2.5.2 多网页模板生成

多网页模板生成算法建立在两个网页的模板生成算法之上。主要过程是选取一个网页作为初始模板，然后根据其他网页逐步调整模板，最后通过统计的方法得到模板，利用此模板生成抽取网页信息的包装器。

首先是初始模板的选取。结合网页聚类的算法，发现对于网页聚类结果簇集合 $C=\{P_0,P_1,\cdots,P_k\}$，满足式(5)的网页 P_i 作为初始模板更为合理。

$$\max_{i=0\to n}\frac{\sum_{j=0\to n}[1-es(p_i,p_j)]}{n} \tag{5}$$

有了初始模板，接下来就是根据其他网页调整和修正该模板。网页的顺序从节点数最多处开始，依次往下，算法的伪代码如下所示：

```
generateTemplate(pageSet, λ)
template ←the page which have the maximum potential template nodes
    Delete the selected template from pageSet
    Sort the pages of pageSet by the number of nodes in descending order
    Mark a integer field appearCount of all nodes in template to 1
    for each page p in pageSet do
    es(template, p)
matchNodesSet=getMatchNodes(template, p)
    for each node pair(nt,np) in matchNodesSet do
        set nt.appearCount=nt.appearCount+1
        alignTemplate(nt,np)
    end
miniCount=ceil((pageSet.count+1)*λ)
discard the nodes whose appearCount is less than miniCount
return template
```

2.6 数据抽取

数据字段的抽取是一个相对简单的过程。只要对要抽取的网页和包装器的相应模板做距离计算，如果模板中的所有必需节点都在最后的映射中，说明该网页满足此包装器，则把与包装器指定的内容节点对应的网页内容部分抽取出来。如果模板中不是所有必需节点都在映射中，则通过距离计算选取最相似的模板抽取网页信息。

3 实验

3.1 评价标准

对于聚类结果精确度的评估标准，采用聚类算法通用的 F-Measure 评估，它结合了信息抽取中的准确率(Precision)和查全率(Recall)的思想：

定义 $C=\{C_1,C_2,\cdots,C_k\}$ 为网页集合 D 的一个聚类结果簇集合，$C^*=\{C_1^*,C_2^*,\cdots,C_k^*\}$ 为网页集合 D 的正确聚类簇结果集合，则簇 j 相对于簇 i 的查全率 $Rec(i,j)$ 可以表示为 $|C_j\cap C_i^*|/|C_i^*|$，簇 j 相对于簇 i 的准确率 $Prec(i,j)$ 可以表示为 $|C_j\cap C_i^*|/|C_j|$，簇 j 相对于簇 i 的 F-Measure 结合这两个值：

$$F_{i,j}=\frac{2\times Prec(i,j)\times rec(i,j)}{Prec(i,j)+rec(i,j)} \tag{6}$$

基于这个公式，聚类结果簇集合 C 的 F-Measure 定义为：

$$F=\sum_{i=1\to n}\frac{|C_i^*|}{|G|}\max_{j=1\to m}F(i,j) \tag{7}$$

F-Measure 的值在 0~1 之间，为 1 时表示完全正确。

3.2 抽取结果分析

本文实验分别从 car.autohome.com、ebay.com、shopping.yahoo.com 三个网站中各选取 500 个网页进行内容抽取，并采用信息抽取工具 RoadRunner 进行对比实验。实验中不断地调整阈值 Ω_g 和 Ω_e，以达到最优的抽取结果，如图 2 所示，当 Ω_g 和 Ω_e 的取值分别为 0.9 和 0.8 时，聚类结果的 F-measure 值达到最优。

从三个网站的信息抽取结果可知，本文基于网页聚类的方法能取得较好的准确率和查全率，平均值分别达到 80.3%和 81.5%。比较 RoadRunner 的抽取结果，平均 67.3%和 66.6%的准确率和查全率，本文提出的方法明显能达到较好的抽取结果，因为 RoadRunner 没有考虑网页文本节点模板，而且对一个页面出现多个数据集的情况不能提取网页的主要内容。分析本文方法对个别网站抽取结果不太理想，是因为网站产品信息列表分布不均匀，主要信息块比较分散，造成准确率和查全率比较低。对于其他大部分主要信息块比较集中的数据提供网站，该方法抽取准确率和查全率在 75%~85%，个别高度模板化的网站甚至可以达到 90%，网页内容抽取实验结果如表 1 所示。

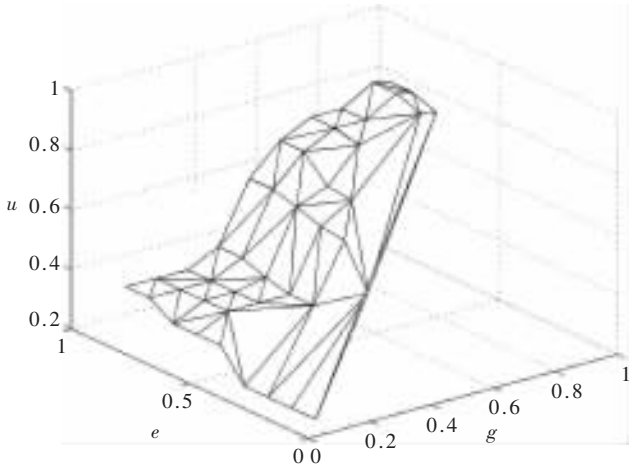

图 2 聚类结果的 F-measure 值

表 1 抽取网页内容的实验结果

网站	本文方法		RoadRunner	
	R/%	P/%	R/%	P/%
ebay.com	74.5	77.3	63.3	60.4
car.autohome.com	89.9	85.7	73.3	71.5
shopping.yahoo.com	79.5	84.4	68.4	70.8
平均值	81.3	82.5	68.3	67.6

本文介绍了一种用于 Web 动态网站的网页聚类方法，利用生成高相似度的网页簇生成高效的包装器，并且成功地用于信息提取，取得了较好的效果，而且与同类技术相比具有算法构造简单、准确率高等优势。该方法适用于信息项内容的结构变化不是很大的 Web 页面，但另一方面，对于信息项的结构变化较大、数据块较多的情况准确率还有待提高。下一步主要工作就是改进抽取模板生成算法，准确识别网页中的主要数据块，提高算法的通用性，以适用于各种类型的网页。

参考文献

[1] CHANG H, KAYED M, GIRGIS R, et al. A survey of web information extraction systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(10): 1411-1428.

[2] RAGGETT D. Clean up your web pages with HP's HTML tidy[J]. Computer Networks and ISDN Systems, 1998(30): 730-732.

[3] LEVENSHTEIN V I. Binary codes capable of correcting deletions, insertions, and reversals[J]. Soviet Physics Doklady, 1996(10): 707-710.

[4] CRESCENZI V, MERIALDO P, MIDDIER P. Clustering web pages based on their structure[J]. Data and Knowledge Engineering Journal, 2005, 54(3): 279-299.

[5] ALVAREZ M, PAN A, RAPOSO J, et al. Extracting lists of data records from semi-structured web pages[J]. Data Knowledge Engineering, 2008, 24(2): 491-509.

[6] CRESEENZI V, MEEEA G, MERIALDO P. RoadRu- nner: Towards automatic data extraction from large websites[C]. In Proceedings of the 27th International Conferenee on Very Large DataBases, Rome, Italy, 2001: 109-118.

(收稿日期：2010-09-26)

作者简介：

邱韬奋，男，1985年生，在读研究生，主要研究方向：数据挖掘、信息抽取。

杨天奇，男，1961年生，博士，教授，主要研究方向：人工智能、数据挖掘、搜索引擎等。