第32卷 第23期 计 算 机 工 程 2006年12月
Vol.32 No.23 Computer Engineering December 2006
·软件技术与数据库· 文章编号：1000—3428(2006)23—0061—03 文献标识码：A 中图分类号：TP311

一种基于同层网页相似性去除网页噪音的方法

袁明轩，张选平，蒋 宇，赵仲孟
(西安交通大学电信学院软件研究所，710049)

摘 要：一个普通的 Web 页面可以被分成信息块和噪音块两部分。基于 web 信息检索的第 1 步就是过滤掉网页中的噪音块。通过网页的特性可以看出，同层网页大多具有相似的显示风格和噪音块。在 VIPS 算法的基础上，该文提出一种基于同层网页相似性的匹配算法，这个算法可以被用来过滤网页中的噪音块。通过实验检测，算法可以达到 95%以上的准确率。

关键词：网页噪音；VIPS 算法；相似树比较

Noise Elimination Method in Web Pages Based on the Similarity of Same Layer Pages

YUAN Mingxuan, ZHANG Xuanping, JIANG Yu, ZHAO Zhongmeng
(Institute of Software, Dept. of Computer Science & Engineering, Xi'an Jiaotong University, Xi'an, 710049)

【Abstract】 A common Web page could be separated into two categories: valuable segments and noise segments. The first step of information retrieval on the Web is to eliminate noise segments or blocks. This paper studies the properties of Web pages and finds out that Web pages with a common URL prefix always have the similar presentation styles and noise segments. Based on vision-based page segmentation (VIPS), it proposes an approximate sub-tree matching algorithm, which could be used to eliminate noise segmentations in a Web page. The implemented algorithm could achieve 95% accurate noise block.

【Key words】 Web page noise; VIPS algorithm; Approximate tree-matching

1 概述

随着网络技术的发展，基于 Web 的信息检索变得越来越重要。但是 Web 数据除了有用的信息之外，还包括一些和网页主体内容关联不大的部分，如网页的页眉、导航栏和著作权脚注等，这些部分被称作噪音块。为了保证针对网页的信息检索的准确性，在对网页进行关键词抽取或主题抽取等操作前，需要将噪音块过滤出去。

人工去除网页噪音块的代价很大，而且易于出错[3]。如何采用自动方式去除网页噪音成为一个研究的热点。文献[1]认为所有的噪音块具有相似的风格，基于此提出了Site Style Tree(SST)的思想，在对网站网页的取样建立SST后，通过SST的测量，得到噪音部分。这种算法的目的是在整个网站中寻找噪音部分。针对找寻一个网页的噪音块，Yin和Lee[4]考虑了网页设计者在设计网页时会将重要的内容放在用户最容易看见的位置，在将网页转换成基本元素以后，利用Page-Ranking的算法计算每一个元素的重要性，由此得到哪些元素在该网页中是重要的。文献[3]针对动态产生的网页使用Hish算法对Augmented Fragment(AF)进行shingles编码，通过编码来检测网页子树的变化情况，由此找出噪音部分。

上述算法能够找到网页中的重要部分，但是那些不重要的部分不一定为噪音块。例如人们访问yahoo新闻时，网页左侧出现一些新闻焦点，虽然它的位置和浏览指南的位置相同，但是它在该网页中并不是噪音块。同时，Yu等提出了Vision-based Page Segmentation (VIPS)[2]，该算法综合考虑了文件对象模型DOM中具有特殊作用的标签(如：P表示一段文字的开始，UL用来表示一个列表)和视觉上的分隔符(如空白区域、字体大小)，将一个网页分割成语义上相近的段落。通过观察，具有相似路径的同层网页在使用VIPS算法进行分割后，噪音块和主题块基本上被分割开来，而且噪音块DOM Tree结构基本上是相似的，只是在颜色或者文字上有些微差别。

针对自动方式去除网页噪音算法的不足，本文在 VIPS 算法对网页分割的基础上，提出了一种基于同层网页相似特性的去噪方法。算法假设同一网站的同一目录下，两个网页的噪音块具有结构和内容的相似性，可以通过相似度匹配的方法，简单而有效地去除噪音块。

为了检验算法的准确性，本文使用了国内比较常用的新闻网站作为测试集，通过测试，算法的准确性可以达到 95%以上。

2 同层网页的相似性

本文算法是基于同层网页具有相似结构这个前提来去除网页噪音。同层网页即在网站的导航结构下同属于同一个节点的子节点的网页。现在的网站设计多采用自动生成程序(如PHP)或模板，这样导致大部分网站的同层网页具有类似的显示效果，而且这些在同层网页中多次重复的模块大多是导航栏、著作权声明等噪音信息。通过网络爬虫分析目标网页中的链接可以获取同层网页。

通过访问 CNN、BBC、搜狐、新浪和中华网等常用的新

作者简介：袁明轩(1980－)，男，硕士生，主研方向：信息检索；张选平，副教授；蒋 宇，硕士生；赵仲孟，副教授

收稿日期：2005-12-07 **E-mail：**fish_yuer@eyou.com

闻网站，西安交通大学、清华大学等国内知名大学的主页和 Amazon、淘宝网、Ebay 等电子商务网站，发现大多数的网页都遵循这个原则。一般地说，网页的 URL(Uniform Resource Locator)越长，其同层网页的噪音相似性越大。

为验证同层网页的相似性，这里通过试验证明。实验中分别选取来自新闻、教育、电子商务以及信息 4 类网站中具有代表性的新浪、西安交通大学主页、淘宝网和证券股票网，在这 4 个网站中随机抽取网页并通过本文算法得到同层网页，然后对这些同层网页的结构进行测试分析，测试结果最后通过视觉效果得到的相似性反映出来。

对同层网页的相似性考察分为 4 部分：页首相似，导航栏相似，页尾相似以及广告部分相似。这里对于每个部分的相似性，根据视觉效果进行评分，评分标准为：每增加一个部分的相似性则相似性分值加 1，即相似性分值介于 0 到 4 之间。验证同层网页的相似性通过相似性度量值反映出来。

相似性度量值定义为

$$\text{相似性度量值}=\frac{\text{网络爬虫得到的同层网页相似性分值总和}}{4\times\text{同层页面数}}\times 100\%$$

通过对结果的分析，实验证明同层网页的结构相似性平均高达 97.3%。实验数据记录如表 1 所示。

表 1 同层网页的相似性

网站名称	样本数量	网络爬虫得到的同层页面数	同层网页相似性分值总和	相似性度量值(%)
新浪	5	24	96	100
西安交通大学网站	5	85	335	98.5
淘宝网	5	41	158	96.3
证券股票网	5	90	339	94.2

3 基于同层网页相似性的去噪方法

3.1 去噪方法

基于同层网页相似性去除噪音的基本思想是：根据同层网页的相似特性，将相似程度超过某一给定阈值的网页部分滤除来得到主体部分。这里考虑到网页表示使用的超文本标签语言(HTML)是一种半结构化语言，其结构信息和字体信息等保存在标签里面，内容信息保存在文本里面。如果两个 HTML相似的话，需要考察其结构相似度($Tag_{similarity}$)和内容相似度($Text_{similairty}$)。相似度($Similarity$)的表述如下：

$$similarity=(Tag_{similarity},Text_{similarity})$$

其中：

$$Tag_{similarity}=\frac{\text{匹配成功的节点数}}{\text{子树的节点总数}},\ Text_{similarity}=\frac{\text{匹配成功的文字长度}}{\text{子树的文字长度}}$$

子树由调用 VIPS 算法切割网页得到。

由于每个网页在设计的过程中，可能会在导航栏、页眉等位置作出一定的修改，反映在 HTML 代码上则是一些表示颜色和字体的属性值有所区别。本文算法中，认为结构相似是最重要的，故允许其在颜色、字体和内容上有一定的不同。为此，定义了最低相似值：(α,β)。

当 $(Tag_{similarity}\geq\alpha)\wedge(Text_{similarity}\geq\beta)$ 时认为子树匹配成功，此时子树是噪音块。

滤除噪音块的过程大致可以分为 3 个步骤：(1)使用网络爬虫获取源网页的同层网页(参见第 2 节)；(2)调用 VIPS 算法将要过滤网页切割成合理数目的子树；(3)对于由 VIPS 算法得到的每一个子树，调用子树匹配算法和同层网页进行匹配，如果计算出的相似度大于预先设定的阈值，就认为是噪音部分，否则就认为是信息部分。

综上所述，滤除噪音块主要采用子树匹配的方法，该方法以基本的字符串匹配算法为基础，加速匹配过程，提高了算法的效率。

3.2 子树匹配算法

子树匹配算法的目的是为了判断输入的两个 DOM Tree 是否存在包含关系。算法的一个输入为通过 VIPS 算法得到的源网页的一棵子树 A，另一个输入为其同层网页树 B。通过匹配算法，可得到 A 是否为 B 的一部分的结论。树与树之间的比较复杂、时间复杂度和空间复杂度都比较高。本文采用将树转化成字符串后进行串比较来加速匹配。具体算法如下：

(1)使用深度优先搜索算法遍历 DOM Tree，将其树状结构转化为节点串。注意在转化的过程中，保存每一个节点的深度信息。

(2)使用子树匹配算法模糊匹配转化过的两个节点串。如果统计出的$Tag_{similarity}$和$Text_{similarity}$超过了阈值，则认为匹配。如果在同层网页节点串的末尾仍然无法获得匹配，则退出，并且返回无法匹配。

(3)由于在转换节点串的过程中，不能保证 DOM Tree 和转换后的节点串是一一对应的关系，因此，需要对两个网页的匹配部分重新进行一次验证，来保证子树匹配算法找到的两个节点串为相似的两棵树。如果检测结果为相似的树状结构，返回 true，否则返回(2)继续进行查找。

子树匹配算法如下：

```
while i<S.length do
  if state=normal then
    if S[i].tagname=T[i].tagname then
      TextMatch(S[i].text,T[i].text,sameTextLength);
    else
       state=s_move;
       Move(S,i);
    end if
  else if state=s_move then
    if S[i].tagname=T[i].tagname then
      TextMatch(S[i].text,T[i].text,sameTextLength);
    else
      if movingstep==MAX then
        Reset(i,j,sameTextLength);
        state=t_move;
      else
        Move(S,i);
      end if
    end if
  end if state=t_move then
    if S[i].tagname=T[i].tagname then
      TextMatch(S[i].text,T[i].text,sameTextLength);
    else
       if movingstep==MAX then
         Reset(i,j,sameTextLength);
         state=normal;
       else
         Move(T,j);
       end if
     end if
  end if
   if j==T.length then
     if(matchTagNum/T.length,sameTextLength/T.text.length)
>(a,b) then
```

```
            return trune;
            else
            if state==t_move or state==s_move then
               Reset (i,j,sameTextLength);
               state=normal;
            else
               i=i-j+1; j=0;
            end if
          end if
        end if
      end while
```

其中 MAX 表示模糊匹配时最大允许滑动过的子树数目。State 用来表示当前工作模式，normal、 s_move 和 t_move 分别表示正常、母串滑动和子串滑动。程序结束后，返回是否匹配的真假值。

由于在将子树转化为节点串的过程中可能将一些不属于同一子树的节点排列成连续的节点串，导致子树匹配算法得到错误的匹配结果。如图 1 所示，A 树和 B 树是完全不同的两棵树。但是，在深度优先搜索后，将得到完全相同的串 ABCDE。因此在子树匹配时，无论要求结果多么精确，都会得到 A 树和 B 树完全相同的结论。

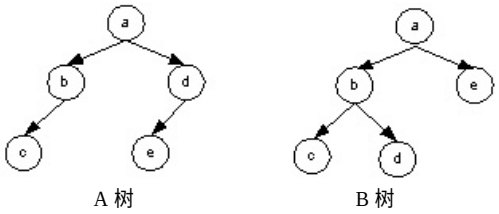


图 1 深度优先搜索结果相同的树

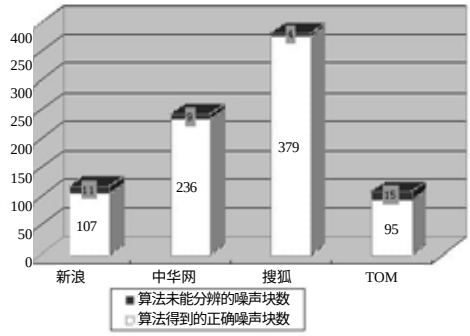
为了解决这个问题，本文在子树匹配后将进行二次结果的验证。假定源网页子树节点串为$A(A_1,A_2,...,A_m)$，通过子树匹配算法获得的同层网页子树节点串为$B(B_1,B_2,...,B_n)$。首先，确定B串中除首节点B1 外的其它所有节点均为首节点的子孙。否则B为来自森林，而非一棵子树。其次，对于每一个匹配到的节点对(A_i,B_i)，比较其相对于根A_1、B_1的深度差。如果每一个匹配到的点的高度差均相等，则两棵树是相同的，否则不相同。例如由图中A树和B树的节点d高度不同可以判断两树不同。

4 实验结果

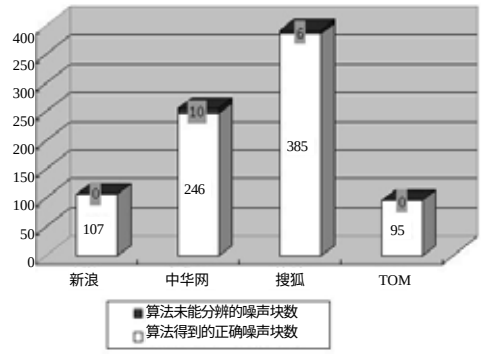
实验选取国内常用的新浪、搜狐、中华网和 TOM 等新闻网站作为实验测试集。从新浪、搜狐、中华网和 TOM 网站中分别选取 15 个网页作为测试用例。当 VIPS 算法中文件相关度 pdoc 为 5(保证网页的噪音部分和信息部分已经分割开)，相似度 (αβ)设定为经验值 (0.8,0.6)时，得到的实验结果如图 2 所示。

观察没能过滤掉的网页噪音块时，发现这些噪音块大多来自于网页的广告，由于网站的拥有者需要各家企业的广告收入，导致在不同的网页上会有不同的动态广告生成，由于广告词不尽相同，而且插入广告的结构一般为 JavaScript 或者图片，结构简单，因此稍有变化就很难被过滤出去。所以对于广告需要单独地分析和处理。

实验中出现了一个错误的过滤，这是由于在使用网络爬虫进行选择时，爬虫选择了该新闻的下一页，也就是说比较的两个网页为同一新闻，使得过滤过程中相关新闻部分被过滤出去，造成错误。但是，由于新闻网页大部分在同一页中显示，这种错误是不常见的。



(a)



(b)

图 2 新浪、中华网、搜狐和 TOM 的测试结果

通过实验结果可以看出，对于大多数网页，算法都可以比较准确地去除噪音块。这里使用两个参数来表示算法的准确性：

$$recall = \frac{\text{算法得到的正确噪音块数}}{\text{网页中的真实噪音块数}} \times 100\% = 95.4\%$$

$$precision = \frac{\text{算法得到的正确噪音块数}}{\text{算法得到的总噪音块数}} \times 100\% = 98.1\%$$

5 结论

在同层网页相似的基础上，本文综合考虑了网页的结构和内容两个方面的相似性关系，提出了相似树比较算法。算法将树的匹配算法转化成为串的匹配问题，简化了问题的处理，提高了算法的效率，降低了时间复杂度。算法期望能够为元搜索引擎的二次处理提供较为准确的主体部分抽取。通过实验，算法的准确率在 95%以上，基本可满足用户的需求。同时，考虑采用 KMP 算法改进本算法，进一步提高效率。

参考文献

1 Yi Lan, Liu Bing . Eliminating Noisy Information in Web Pages for Data Mining[C]. Proc. of International Conference on Knowledge Discovery and Data Mining, Washington D. C., USA, 2003-08.

2 Yu Shipeng, Cai Deng, Wen Jirong, et al. Improving Pseudo-relevance Feedback in Web Information Retrieval Using Web Page Segmentation[C]. Proceedings of WWW2003, Budapest, Hungary, 2003: 11-18.

3 Ramaswamy L, Iyengar A, Liu Ling, et al. Automatic Detection of Fragments in Dynamically Generated Web Pages[C]. Proceedings of WWW2004, New York, USA, 2004: 443-454.

4 Yin Jirong, Lee W S. Using Link Analysis to Improve Layout on Mobile Devices[C]. Proceedings of WWW2004, New York, USA, 2004: 338-344.