Hans Journal of Data Mining 数据挖掘, 2015, 5(4), 69-74
Published Online October 2015 in Hans. http://www.hanspub.org/journal/hjdm
http://dx.doi.org/10.12677/hjdm.2015.54010

Hans汉斯

Research and Implementation of Text Information Extraction Based on WEB

Sanxiang Liu

Zhaoqing Industry & Trade Vocational School, Zhaoqing Guangdong
Email: liusx333@163.com

Received: Oct. 25th, 2015; accepted: Nov. 19th, 2015; published: Nov. 26th, 2015

Abstract

In this paper, based on the theory and method of traditional information extraction, a method of Web Text Extraction Based on XML features is realized. The characteristics of general web pages are studied. A method of web page extraction based on XML tag feature is implemented. The HTML pages are standardized. The XML language is converted into XML language. According to the features of XML language, the internal language is transformed from GB to UTF, and then the standard is also extracted.

Keywords

Internet, Information Extraction, HTML, XML, Text Information Extraction

基于WEB网页文本信息抽取研究与实现

刘三星

肇庆市工业贸易学校，广东 肇庆
Email: liusx333@163.com

收稿日期：2015年10月25日；录用日期：2015年11月19日；发布日期：2015年11月26日

摘 要

本文以传统的信息抽取理论和方法为基础，实现了一种基于XML特征的网页文本抽取方法。研究了一般

文章引用: 刘三星. 基于 WEB 网页文本信息抽取研究与实现[J]. 数据挖掘, 2015, 5(4): 69-74.
http://dx.doi.org/10.12677/hjdm.2015.54010

网页的特征，实现了一种基于XML标签特征的网页提取方法，对网页进行HTML页面标准化，将其转成XML语言，并且根据XML语言的特点，对其内部语言进行转化，从GB转为UTF，并对其进行标准化，然后通过熟悉XML标签的各种特性，对网页文本根据标签进行抽取。

关键词

互联网，信息抽取，HTML，XML，文本信息抽取

1. 引言

Web网已经成为一个巨大的信息源，数据量急剧地膨胀，Web网也成为人们获取信息的重要来源[1]。然而 Web 页面中存在着大量的 HTML 格式的无结构数据和少量 XML 格式的半结构数据[2]。网页抽取也成为信息搜索(Information Search)、数据挖掘(Data Mining)、机器翻译(Machine Translation)和文本摘要(Text Digest)等 Web 信息处理的基础。

Web 信息抽取(Web Information Extraction，简称 WIE)是指：给出属于同一类型的若干样本网页。找出它们的源数据集的嵌套结构，并将源数据集从网页中抽取出来。即通过对原文档信息内容和结构的分析，抽取出有意义的部分，生成结构化的有价值的信息。

Web 信息抽取渐渐成为一个崭新而热门的课题，从互联网资源中抽取数据的传统方法就是编写特定的程序，这种程序被称为“Wrapper”。Wrapper 是一个能够将基于 HTML 描述的 Web 网页内容转换为按照某种结构化描述的数据集合(例如 XML 数据、关系数据库)的软件程序。它由信息抽取所需的信息识别与结构影射知识和应用这种抽取知识的处理程序组成。根据各种工具用于产生 Wrapper 而采取的不同技术，目前的 Web 数据抽取工具可分为六种：Wrapper 开发语言，可感知 HTML 的工具，基于 NLP 的工具，Wrapper 归纳工具，基于建模的工具，基于语义的工具[1]。

本文从理论上分析网页文本信息抽取的方法及流程，具体阐述了网页文本信息抽取的理论和方法，以当当网页文本信息抽取为例，介绍了基于标签的信息抽取系统的概述，同时阐明了具体的过程和模块，给出该抽取实现方法的步骤以及实现的某些核心代码，分析此方法的优点和可以进一步改进的地方，并就其意义和所需进一步思考的地方进行了阐述。

2. Web 网页文本信息抽取的流程和原理

无论挖掘的目的是什么，都可以把 Web 文本挖掘的一般处理过程用图 1 来概括。

目前解决网页信息抽取问题比较典型的方法有：基于自然语言处理(NLP)方式的信息抽取；基于包装器归纳方式的信息抽取；基于 ontology 方式的信息抽取；基于 HTMI 结构的信息抽取；基于 web 查询的信息抽取等[2]。

基于 web 查询的信息抽取：是利用数据库技术在对互联网的网上数据进行管理和查询，将 Web 信息抽取转化成运用标准的 Web 查询语言对 Web 页面文档进行查询，具有很强的通用性。采用这种技术的系统有：Web-OQL 以及自主开发的原型系统 PQAgent [3]。

把网页标准化成为 HTML，然后将其转化成为 XML，根据 XML 语言进行抽取信息。

3. 网页文本信息抽取设计

3.1. 网页文本信息抽取设计方案

多数 Web 文档都是把标记和文本按照 HTML 的定义联在一起的。标记包括“<”和“>”，在“<”

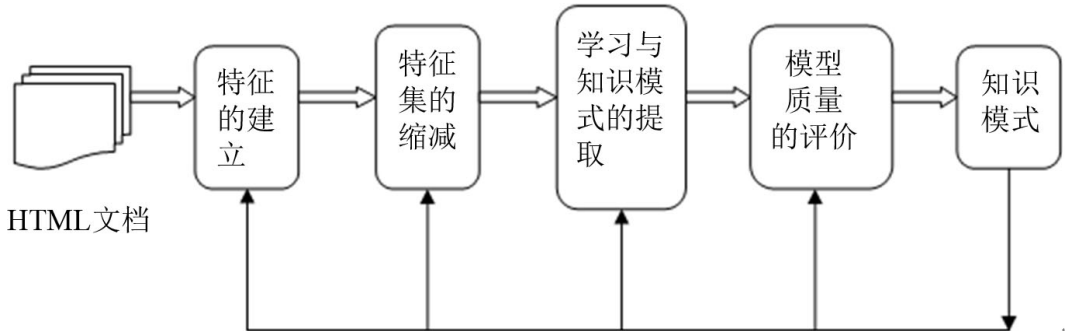

Figure 1. General processing procedure of Internet text data mining
图 1. Internet 上文本数据挖掘的一般处理过程

和“>”之间为标记名称。标记经常成对出现，分别为开始标记和结束标记。开始标记不以“/”开头，而结束标记以“/”开头。

现在，每个页面都是由标记“DIV”有良好的布局特性，格式复杂的页面往往都采用“DIV”标记来进行页面格式的布局。因此，可利用“DIV”标记对页面进行解析。网页文本信息抽取的基本流程如图 2 所示。

3.2. 网页文本信息抽取过程

从 HTML 到 XML 的转化如图 3 示。

在互联网上，找到部分标签的正则表达式如下：

```
String regEx_script= "<[\\s]*?script[^>]*?>[\\s\\S]*?<[\\s]*?\\/[\\s]*?script[\\s]*?>";
//定义 script 的正则表达式{或<script[^>]*?>[\\s\\S]*?<\\/script> }
String regEx_style= "<[\\s]*?style[^>]*?>[\\s\\S]*?<[\\s]*?\\/[\\s]*?style[\\s]*?>";
//定义 style 的正则表达式{或<style[^>]*?>[\\s\\S]*?<\\/style>
    过滤部分标签：
        p_script=Pattern.compile(regEx_script,Pattern.CASE_INSENSITIVE);
        m_script = p_script.matcher(HTMLStr);
        HTMLStr = m_script.replaceAll(""); //过滤 script 标签
        p_style = Pattern.compile(regEx_style,Pattern.CASE_INSENSITIVE);
        m_style = p_style.matcher(HTMLStr);
        HTMLStr = m_style.replaceAll(""); //过滤 style 标签
p_HTML = Pattern.compile(regEx_doc,Pattern.CASE_INSENSITIVE);
        GB 转化为 UTF：
InputStreamReader reader = new InputStreamReader(in, "GB2312");
//以 GB2312 编码读入文件
FileOutputStream out = new FileOutputStream(tmpFile);
//将文件转化为字符流
OutputStreamWriter writer = new OutputStreamWriter(out, "UTF-8");
//目标编码为 UTF-8
writer.write("<?XML version=\"1.0\" encoding=\"utf-8\"?>\n");
            char[] buffer = new char[10240];
```

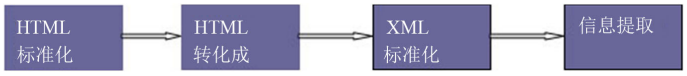

Figure 2. Basic process of Web text information extraction
图 2. 网页文本信息抽取的基本流程

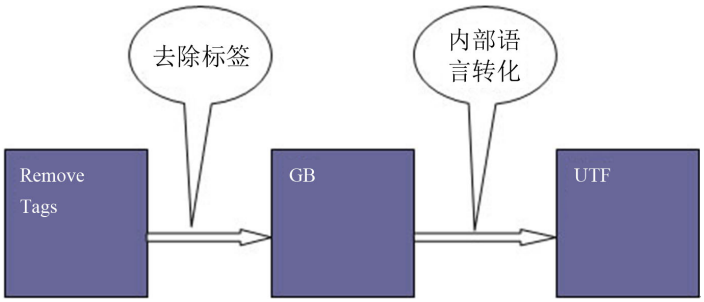

Figure 3. Conversion process from HTML to XML
图 3. HTML 到 XML 转化过程

其中，如果不进行 GB 至 UTF 编码的转化，就有可能出现乱码。出现乱码的原因是多样的，如各个模块编码不一样就会引起这样的情况。GB 即“国标”的汉语拼音缩写，为中华人民共和国国家标准的意思。国标编码就是中华人民共和国信息交换汉字编码标准，在此标准中制定了每一个汉字及非汉字符号的编码。文档的当前编码不能正确保存文档内所有字符，就需要改为 UTF 或其它支持本文文档中的特殊字符(如'\0'或'/'等)的编码[4]。

3.3. 信息抽取部分实现代码

```
public static void inspect(Element element){
        if(!element.isRootElement()){
            System.out.println();
        }
        String qualifiedName=element.getQualifiedName();
    if(qualifiedName.equals("a"))
        qualifiedName = "超链接";
    else if(qualifiedName.equals("title"))
        qualifiedName = "网页标题";
    else if(qualifiedName.equals("img"))
        qualifiedName = "图片链接";
    else if(qualifiedName.equals("html")
                || qualifiedName.equals("head")
                || qualifiedName.equals("div")
                || qualifiedName.equals("table")
                || qualifiedName.equals("center")
                || qualifiedName.equals("body"))
        return;
     // System.out.println(qualifiedName+":"+element.getText());
        Namespace namespace=element.getNamespace();
```

```
            if(namespace!=Namespace.NO_NAMESPACE){
                String localName=element.getName();
                String uri=element.getNamespaceURI();
                String prefix=element.getNamespacePrefix();
               // System.out.println("      Local name: "+localName);
               if(!"".equals(prefix)){
                        // System.out.println("      Namespace prefix: "+prefix);
                }
            }
        String url=args[0];
            try{
                SAXBuilder parser=new SAXBuilder();
                Document document=parser.build(url);
                process(document.getRootElement());
            }catch(JDOMException e){
                System.out.println(url+" is not well-formed.");
            }catch(IOException e){
                System.out.println("Due to an IOException,the parser could not encode "+url);
            }
    }
}
```

4. 信息抽取实现

以当当图书-求医不如求己.HTML 和卓越亚马逊.HTML 为例，如图 4 所示。(其中，标准化 HTML 中，图片存在相应名字的文件夹中)。

卓越亚马逊：

HTML 标准化转化成 XML 页面，根据 XML 语言中的部分标签提取文本信息，如图 5 结果。

Figure 4. Web page of Joyo.com
图 4. 卓越网页面

Figure 5. Extraction results of Dangdang
图 5. 当当网抽取结果

5. 总结

实现一种基于 XML 语言标签特征的抽取实现方法，详细介绍其原理和具体过程；对其进行测试和评估，给出测试结果，并且演示部分的网页文本信息抽取的结果；对信息抽取方法进行思考和总结。

参考文献 (References)

[1] 陶庆, 刘峰. Web 数据挖掘在电子商务中的应用研究[J]. 电脑知识与技术, 2008(12): 415-416.

[2] Chang, C.H., Kayed, M., Girgis, M.R. and Shaalan, K.F. (2006) A Survey of Web Information Extraction Systems. *IEEE Transactions on Knowledge and Data Engineering*, **18**, 1411-1428. http://dx.doi.org/10.1109/TKDE.2006.152

[3] 毕蕾, 沈洁, 徐法艳, 魏榴花, 朱燕, 孙荣霜. 领域本体指导的 Web 商品信息抽取[J]. 计算机工程与设计, 2008, 29(24): 6393-6396.

[4] Laender, A.H.F., Ribeiro-Neto, B.A., da Silva, A.S. and Teixeira, J.S. (2002) A Brief Survey of Web Data Extraction Tools. Federal University of Minas Gerais, Belo Horizonte.