

基于XML和N层VSM的Web信息检索

张冉,卡米力·毛依丁

(新疆大学信息工程学院 计算机系,新疆 乌鲁木齐 830046)

摘要:基于XML文档格式良好、层次清晰,可以方便地操纵、分析其结构的特点。文中在将Web上的HTML文档转化为XML文档的基础上,通过Java中的DOM树,分析文档的层次结构。把文档分为层次化的文本段,对传统的VSM算法进行改进,把每个文本段转换为空间向量,实现了N层VSM算法,通过试验证明,改进后算法的查全率和查准率都要优于传统的VSM算法。

关键词:XML; XHTML; N层向量空间模型; 查全率; 查准率

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2006)05-0056-03

Web Information Retrieval Based on XML and N-level VSM

ZHANG Ran, Kamil·Moydin

(Computer Dept., College of Infor. Sci. and Eng., Xinjiang Univ., Urumqi 830046, China)

Abstract: XML documents have well form, clear levels and analyses the structure easily. Convert HTML documents on Web into XML document, so can use DOM tree in Java to analyse the hierarchy of the documents. The documents can be divided into N level text paragraphs' content, which are represented by index term vectors. Using this method improve traditional vector space model, the N level VSM is achieved. And proved by the experiment, both recall and precision of the N level VSM are performing well than the traditional VSM.

Key words: XML; XHTML; N-level VSM; recall; precision

0 前言

在一篇文档中,出现在不同位置的词,代表文档的能力会有所不同。传统的向量空间模型在检索文档的时候,忽略了文档本身的层次结构。于是,使用XML将Web上的文档标准化^[1]。分析其层次结构,把文档的层次信息结合到文档的检索中去。

1 XML

XML^[2]是由W3C发布的一种新标准,它是SGML的一个简化子集,将SGML丰富的功能和HTML的易用性结合起来,以一种开放的、自我描述的方式定义数据结构。XML文档由标记和字符数据组成,如图1所示。通过DTD或Schema使XML文档结构化,这样很容易验证文档数据的合法性,容易提取(查询)文档中的数据。可以利用CSS或XSL在浏览器中实现同一XML文档的多种显示形式,也可方便地将XML文档译为HTML文档或者不同标记表示的XML文档。XML将网络信息标准化,使开发者和计算机易于辨认信息,能创建不依赖于平台、语言或者格式有开放的开放数据。下面介绍文中用到的

XML标准体系中的一些标准。

```
<paper>
  <title>XML and Java: a marriage made in heaven
</title>
  <preamble>
    <author>John Hunt</author>
    <abstract>What is the ... </abstract>
  </preamble>
  <body>
    <introduction>This paper ... </introduction>
    ...
  </body>
</paper>
```

图1 XML文档

1.1 XHTML

根据用户需求,抽取相关HTML页面上的信息。当前许多Web站点上的HTML代码并不是格式完整的,换句话说HTML对格式完整并没有什么严格要求,解析HTML的浏览器如IE或Netscape都可以容忍一定格式上的缺陷。因此,首先要把这种格式非良好的HTML文档转变成格式良好的XML文档。其次通过分析XML文档提取用户所需的信息。

XHTML是一个与XML兼容的HTML版本,包含所有的HTML元素和属性,XHTML文档有良好的规则。

收稿日期:2005-09-07

作者简介:张冉(1981-),女,新疆乌鲁木齐人,硕士研究生,研究方向为网络信息检索;卡米力·毛依丁,硕士生导师,副教授,主要研究方向为网络信息安全。

可以利用 W3C 站点上的 HTML tidy 工具,实现自动转化。转化命令为:

tidy - raw - asxml -f 错误信息文件名 输入文件名
> 输出文件名

1.2 DOM(Document Object Model)

XML 文档在内存中的保存形式类似于数据结构中的树型结构,树型结构的节点对应着文档中的元素以及元素属性,同时节点还保存着元素或属性的值,即人们所关心的 XML 文档内容。

DOM 接受 XML 文档,构建表示 XML 文档的树状结构,如图 2 所示。可得到树中的任何元素(用节点对象表示)及其特性、子节点、值等。这些通过一组标准的规范实现。就是说,W3C 的 DOM 规范实际上是一组接口,这些接口指定了 DOM API 的各种元素应该做什么。然后使用特定的语言(如 Java)实现接口以创建具体的实现^[3]。

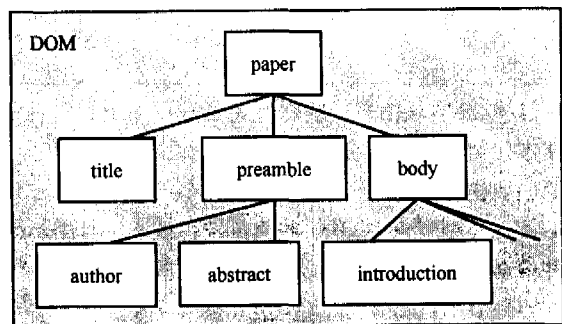


图 2 构建 DOM

2 N 层向量空间模型

向量空间模型在传统信息检索模型中,使用的比较广泛。但是,在检索的时候忽略了文档的层次结构,于是提出了基于文档层次结构的 N 层向量空间模型。

2.1 向量空间模型

向量空间模型^[4]引入了线性代数的知识,若干独立的词项被选作索引项,用检索项的向量空间表示用户的检索要求和数据库文档信息。每一篇文档都被映射成多维向量空间中的一个向量,对于所有的文档类和未知文档,都可用此空间中的向量 $D_j(d_{1,j}, d_{2,j}, \dots, d_{t,j})$ 来表示(如图 3 所示)。其中, t 是系统中所有索引项的个数。 $d_{i,j}$ 为索引项 k_i 在文档 d_j 中对应的权值,用以刻画该索引项在描述此文档内容时的重要程度,使用公式(1)进行计算:

$$w_{ik} = tf_{ik} \times idf_k = tf_i \times (\log_2(N/n_k) + 1) \quad (1)$$

从而将文档信息的表示和匹配问题转化为向量空间中向量的表示和匹配问题来处理。查询中的索引项也是有重权的,表示为 $Q = (q_1, q_2, \dots, q_t)$, 这样,查询式和文档都表示为向量,由此可以通过计算向量 D_i 和 Q 的相似度来评价文档和用户查询的相似程度。一般使用余弦公式来度量向量的相似度,如公式(2)所示:

$$\text{sim}(D_i, Q) = \frac{\sum_{k=1}^t (d_{ik} \cdot q_k)}{\sqrt{\sum_{k=1}^t d_{ik}^2 \cdot \sum_{k=1}^t q_k^2}} \quad (2)$$

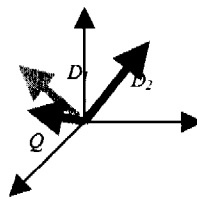


图 3 文献中的空间向量

2.2 N 层向量空间模型

传统的向量空间模型忽略了文档的层次性,把每一个文本段都同等看待。但是,在信息抽取以及查询匹配过程中,同一个关键词在文档中的不同位置,它所能表达文档内容的能力也是有差别的。比如,在一个文档集中有 3 个文档 d_1, d_2 , 和 d_3 , 这 3 个文档中都包含特征项 t , 且 t 在这 3 个文档中出现次数都为 k 次,但是在文档 d_1 中, t 是被包含在标题中;在文档 d_2 中, t 是被包含在摘要中;在文档 d_3 中, t 是被包含在正文中,运用传统的信息搜索引擎则会认为特征项 t 表达这 3 个文档的能力完全相同。而事实上出现在标题中的特征项要比出现在摘要中的特征项更能确切代表文档的内容,同样出现在摘要中的特征项也要比出现在正文中的特征项更能代表文档的内容。

于是 N 层向量空间模型表述为^[5]:模型将一个文档从组织结构上划分为 N 层(N 个文本段),基于每层的文本段内容建立相应的文本特征项向量以及文本权值向量,这样,对于这个文档进行 N 层划分所得到的向量空间模型就成为 N 层向量空间模型。文档表示为树后,将其分为 N 层,给每个节点编号,如图 4 所示。然后选择其中的叶子节点作为文本段,并记录该叶子节点的层数。

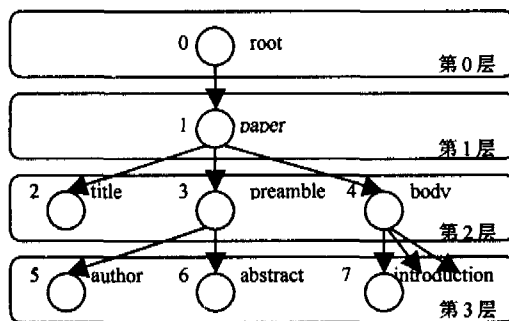


图 4 XML 文档树分层

这样,在查询文档的相似度时,先把查询向量和文档的各文本段相似度用公式(2)计算出来。然后根据各文本段的层数对这些相似度加权求平均数,得到改进的相似度计算公式:

$$\text{sim}(Q, D_i) = \sum_{j=1}^M \text{sim}(Q, d_{ji}) \cdot f_{ji} \quad (3)$$

其中, f_{ji} 为叶子节点,即文本段所在层加权,把根节点看作 1,使文本段所在层加权 = 父节点层权值/兄弟节点个数。如图 4 中 title 的层权值为 1/3,author 的层权值为 1/6。

3 系统设计

从网上获取 1000 篇有关计算机方面的文档,采用

《计算机词汇专用字典》4750 个流行的计算机术语词条，建立特征项库，在 PIV 2.0G，256MB 内存的机器上实现本系统的试验。

3.1 系统结构

整个系统的流程图如图 5 所示，分为用户界面、相似度计算和预处理模块。图中用虚线标出的预处理模块部分是本系统的核心部分。

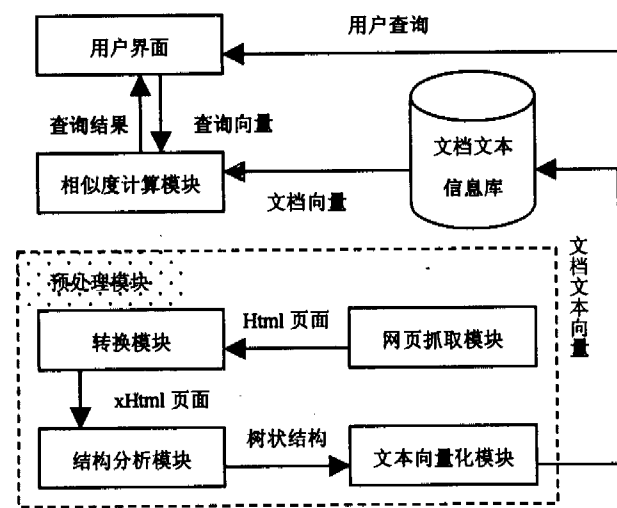


图 5 系统流程图

进入用户界面进行查询，然后通过相似度计算模块从文档文本信息库中读出的文档文本向量，将分层的文档向量与用户的查询向量进行相似度计算，最后把查询结果返回到用户界面。其中，文档文本信息库中的信息由预处理模块实现的，其工作流程如下：通过网页抓取模块获得 Web 上的 HTML 页面，通过转换模块将其转换为 XHTML 页面，标记索引后将其送交结构分析模块，分析出其树状结构，标记层数，将其逐层向量化。最后把整个文档的向量存入文档信息库。

3.2 关键环节

系统中较关键的是预处理模块中树状结构的分析，也就是对 XML 文档进行分析，获取文档的文本段。可使用 Java 读取 XML 文档，对 DOM 树进行分析。XML 文档有很多节点类型，部分类型如表 1 所示。利用这些节点类型和节点名，可得到文档的 N 层文本段。通过 Document 类型的节点找到文档的 root 节点，然后调用 getChildren() 方法将文本插入数据库；getChildren() 方法算法如下：

```
void getChildren(Element elem, int depth){//得到下一层节点信息
    children = elem.getChildNodes();//得到 elem 的下一层节点链表
    for (i=0;i<children.getLength();i++){
        child = children.item(i); //读取一个节点
        type = child.getNodeType(); //读取节点类型
        if (type == ELEMENT){ //如果节点类型是 element
            elemName = child.getNodeName(); //读取节点名称
```

```
insert(elemName, depth + 1);
//将 element 节点名称和所在层数插入数据库
if (child.hasChildNodes()) //是否有下一层节点
    getChildren((Element) child, depth + 1); //递归调用
getChildren()得到下一层节点信息
}
if (type == TEXT){ //如果节点类型是 text
    nodeValue = child.getNodeValue(); //读取节点的值
    if (nodeValue.isVisible()) //如果 text 节点值是可见的
        insert(textValue, depth + 1); //将 text 节点值和所在层数
        插入数据库
    }
}
```

表 1 节点的类型和属性(部分)

节点类型	节点名	节点值
Element	tagName	null
Text	# text	正文节点的内容
Document	# document	null
.....

这样，文本段信息就被保存在了数据库中。然后对这些文本段信息构造特征向量，按照公式计算相似度即可。这其中要说明的就是(算法中划横线的方法)，在 XML 定义的 text 类型节点中，有一些值是空白或不可见的。因为这些值对于检索是没有意义的，所以这些值是被忽略的。

4 结果及实现

评价信息检索算法性能主要指标为查准率和查全率。查准率是指检索到的相关文档数与检索到的所有文档数的比值，它表示检索到的文档中相关文档的比例；查全率是指检索到的相关文档数与总的相关文档的比值，它表示检索到的相关文档的比例。表 2 是试验得到的传统模型与 N 层模型的查全率与查准率比较结果。从表中可以得知，N 层向量空间模型在查全率和查准率上都要好于传统向量空间模型。

表 2 结果比较

	查准率	查全率
传统模型	77.54%	29.37%
N 层模型	92.33%	49.67%

参考文献：

[1] 陈玉芳,葛燧和.一个基于 XML 的 web 数据收集模型的研究[J].计算机工程与应用,2004,40(10):150-152.
[2] McLaughlin B.Java 与 XML(第 2 版)[M].刘基诚译.北京:中国电力出版社,2004.
[3] Hunt J, Loftus C.精通 J2EE-JAVA 企业级应用[M].周立斌,杨飞译.北京:清华大学出版社,2004.
[4] Baeza-Yates R, Ribeiro-Neto B.现代信息检索(英文版)[M].北京:机械工业出版社,2005.
[5] 陈治平,林亚平,童调生.基于 N 层向量空间模型的信息检索算法[J].计算机研究与发展,2002,39(10):1233-1237.