支持 XML 信息检索的索引技术*

宋　玲[1,2]，马　军[1]，郭家义[3]

(1. 山东大学 计算机科学与技术学院，山东 济南 250100; 2. 山东建筑工程学院 计算机科学与技术系，山东 济南 250014; 3. 中国科学院 文献情报中心，北京 100080)

摘　要: 作为互联网的新技术,XML 已经渗透到了 Internet 的很多领域,结构文档 XML 的信息交换、提取、处理、查询的研究也日益受到重视。目前,已经提出了许多面向 XML 的查询语言,这些查询语言一般基于路径和树模式。从信息检索理论与技术角度出发,探讨 XML 文档的索引技术,以期达到内容和结构的双重检索。

关键词: XML; 索引; 信息检索; 查询

中图法分类号: TP335　　**文献标识码**: A　　**文章编号**: 1001-3695(2005)03-0031-03

An Index Technology for XML-based Web Information Retrieval

SONG Ling[1,2], MA Jun[1], GUO Jia-yi[3]

(1. *School of Computer Science & Technology, Shandong University, Jinan Shandong* 250100, *China*; 2. *Dept. of Computer Science & Technology, Shandong Institute of Architecture & Engineering, Jinan Shandong* 250014; 3. *Library of Chinese Academy of Sciences, Beijing* 100080, *China*)

Abstract: As a new technology, XML language has become popular in Internet. More and more researches on information exchange, information extraction, information processing and information retrieval based on XML are paid great attention to. Recently many query language have been used in search. These query language commonly based on path and tree pattern. The paper explores an index method for XML files based on information theory and technology which can implement retrieval both on context and structure.

Key words: XML; Index; Information Retrieval; Query

1　引言

目前, Web 的构造语言主要是非结构化的 HTML 语言,网上信息检索主要是依靠搜索引擎,搜索引擎多是基于关键词的全文检索,搜索结果中常常包含了许多冗余信息,而一些有用的相关信息却检索不出来。例如,用 Google 查找电影“Green tea”,如果以“Green Tea”作为关键词进行检索,你会得到 Green Tea Gum, Green Tea Health Benefit, Green Tea Press, Film-green Tea 等 1 600 000 项查询结果。再例如,一个用户希望了解动态规划算法,虽然查到了有关算法的很多文档,但他还不得不浏览整个文档,查找动态规划这一小部分,这些都大大影响了检索效率。XML 作为 SGML 的子集,为 Web 简化提供了一种定义和描述结构数据的方法,如果上述电影、文献、资料等使用 XML 文档描述,那么检索的查全率和查准率都将大大提高。有专家预测, XML 文档将成为 Web 资源的重要组成部分。XML 可用于建立多层 Web 应用、集成不同的信息源、根据用户个性化的特点检索结果的多种显示等。这一切都说明, XML 为智能代理、人工智能、数据挖掘等技术在信息检索领域的应用开辟了广阔的天地, XML 将使信息检索系统更为智能和准确。在现有的 XML 信息检索研究中,主要从以下几个方面来探讨信息查询:

收稿日期: 2004-04-22; 修返日期: 2004-06-16

基金项目: 山东省自然科学基金资助项目(Z99G01); 国家“863-306”主题基金资助项目(863-306-ZT06-01-4)

(1) 对目前的一些基于 XML 文档的查询语言如 XML-QL, Xpath, Xquery, QUILT 进行扩展和开发,使之支持检索结果排序[1,3~5,13,15],支持传统的关键词查询[13]以及异构数据源的查询[14]。这种方式仅限于从查询语言的角度来考虑信息检索。

(2) 从数据库的角度来存储和检索 XML 文档,借助于标准数据库的查询语言,检索结果能够准确地满足查询条件[6,10]。这种检索模式实际上是对数据库的检索,体现不出 XML 文档检索的意义,无法做到相关查询,另外作为一个检索系统,索引及查询结构排序都是必需的。

(3) 基于信息检索理论,对用户提问与 XML 文档之间计算相似度,依此来返回检索结果。这需要对 XML 文档进行解析,形成文档树,根据文档树与原始结构化文档的对应关系,对节点进行索引,形成倒排文档,将结构检索和语义检索相结合,实现检索过程中的语义扩展、面向 XML 文档的概念检索[2,4,9]、聚类等[2,12]。

优化的索引机制是建立检索系统的基础上,因此,本文将针对 XML 文档的索引机制(包括结构和内容)进行研究。本文描述了信息检索相关的理论、支持 XML 信息检索的索引结构所需要解决的问题和相关的工作,探讨了基于 XML 文档的索引技术,给出了一个完整的索引方法,包括计算 XML 文档树中的词汇权重,从词汇树到汇总树的转换,确定索引词条,建立倒排文档。

2　问题描述

2.1　信息检索理论

信息检索的基本原理是对信息集合与需求集合的匹配与

选择,形式化描述为一个四元组: $S = (D, T, Q, \rho)$。其中, $D = (D_1, D_2, \cdots, D_n)$, $T = (T_1, T_2, \cdots, T_m)$, $Q = (Q_1, Q_2, \cdots, Q_l)$, ρ $Q \times D \rightarrow R$。式中 D 表示某系统中经过标引的文献集合, T 和 Q 分别代表标引词集合和检索提问词集合, ρ为匹配函数,其函数值集合 R 为检索结果。

完成信息检索需要三个过程: ①信息检索系统对信息源中的所有文档进行标引,首先从文档中抽取能反映其主题内容的特征词(索引词),然后对特征词进行索引,形成索引文档。其中,抽取特征词括自动切词、去除停用词(Stopwords)、词干提取(Stemming)、词条权重计算、阈值筛选等技术。②将用户基于自然语言的检索提问分解为多查询关键词并进行语义扩充。③根据处理后的查询关键词在索引表中进行查询,同时完成文档与查询之间的相关度评价,对将要输出的结果进行排序,实现某种用户相关性反馈机制。

2.2 支持 XML 信息检索的索引结构

构造支持 XML 信息检索系统的索引结构,首先要面临的一个问题是如何抽取索引词,用于表示文档以及生成文档的索引表。我们知道,一篇 XML 文献的信息是按层次化的树型结构组织的,而一篇文献由若干章节组成,每一章节由若干段组成,经过 XML 分析器的分析之后,信息被转换成一棵对象节点树。在这棵树中,由一个根节点代表文献,其他的章、段落都是根节点的后代节点,这些节点称为元素,叶节点表示段落(图 1),基于 XML 文档的搜索可以得到部分段落。那么从文档中抽取特征词的问题就转换为从这些节点中抽取特征词的问题(图 2)。

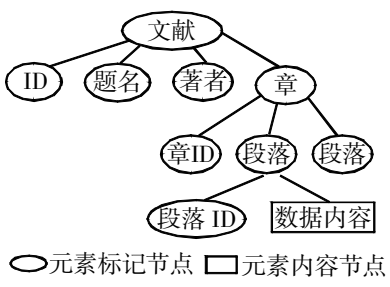

图 1 XML 文档树

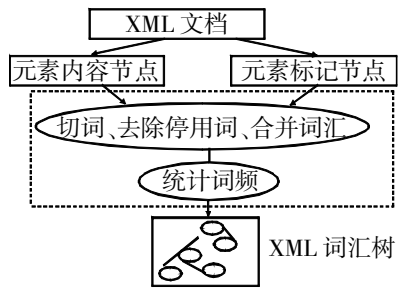

图 2 XML 词汇树

在传统的信息检索中,词汇索引(Term Index)是一个三元组 < *term, document_ID, frequency* >,词汇权重的计算采用常用的 TFIDF 方法。但在 XML 文档中,基于节点元素文本内容的索引就需增加结构信息 < *term, frequency, position* >,其中, *Position* 是结构信息,以图 1 为例, *Position* 的结构为(document_ID, section_ID, paragraph_ID),更详细一些,加上 line_ID,为了满足更具体的检索要求,希望信息粒度较小,还可通过词汇的具体偏移量更详细地定位词汇的位置,如 < information, 4, < 2, 7, 5, 9, 3 > >,表示词汇 Information 在 2 号文献中出现了四次,其中一次出现在第七章第五段第九行第三个词汇。词汇权重的计算要复杂一些,应根据节点的不同位置分别计算,其倒排文件(Inverted File)是一个二元组 < *term, position* >。

基于 XML 文档的检索系统根据检索单位划分有两种: ①以文档为检索单位; ②以文档中被标记的元素(节点)作为检索单位。前者返回的结果是符合用户查询条件的一个文档集,而后者返回的结果是文档中的部分数据(如某个段落)。

基于 XML 文档的检索是实现内容和结构的双重检索,可用以下公式表示[8]:

$$XML: Q \rightarrow M = \mathrm{Sort}(\pi\sigma_{similarity(query, <structure, content>) \geq t})$$

我们将这种把文档的文本内容与文档的结构结合起来的检索模型称为结构化文本检索模型。基于邻节点的模型就是这种结构化文本检索模型。这种模型是一种允许在相同的文档上定义独立的分层索引结构的模型。每个索引结构是一个严格的层次结构,由章、段落、行组成(图 1),其中每个节点都与一个文本区域相关,两个不同的层次结构可能涉及到两个重叠的文本区域。图 3 给出了词汇 Information 的倒排链表,该倒排链表的项列出了文档中所有出现词汇 Information 的位置。

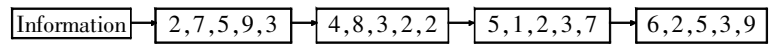

图 3 倒排链表

2.3 相关的工作

目前,研究 XML 文档中的索引机制的理论和实践很多,如 XRS 是基于 BUS(Bottom Up Scheme)技术以文档的最低结构层次上的标记元素作为索引项。Lore 是斯坦福大学开发的 XML 文档数据库管理系统,它支持多索引技术和相似查找技术。伯克利大学的 Xset 系统中设计的索引综合了路径信息,斯坦福大学的 DataGuide 系统实现了任意结构的半结构数据的索引。该索引是关于半结构数据的路径信息的一个精确简洁的索引结构,可有效地用来辅助建立查询和进行查询优化。

文献[2] 介绍了 BUS(Bottom Up Scheme)索引技术,索引只在文档的叶节点上建立,非叶节点的权重信息在检索时动态获得。文档树中的每个节点 GID 由四部分构成: ①文档编号(DID),用于唯一的标志元素所在的文档; ② 按照树型结构上的节点顺序(从上到下,从左至右)给每一个实节点赋予一个单一识别号(UID); ③元素在文档树中的层次(Level) ——索引层(Index Level),具有文本节点的元素所在的层次,是实际进行索引的层次; ④在整个树型结构中的元素类型编号。索引项的组成为(Term, Frequency, GID)。

文献[11] 将 XML 文档看作带有附加标记的文本文档,对 XML 文档的内容和路径进行索引,并借助已有的信息检索技术实现内容与结构的检索。

文献[9] 提出了一个层次索引机制的算法,对所有节点进行索引,但为了避免词汇的重复索引提出了一个传播和修剪算法(Propagation and Pruning Mechanism)来选择索引词。从目前 XML 文档的索引技术来看,导致信息检索存在以下几个方面的不足: ①只对叶节点进行索引,忽视了非叶节点(内点)中所包含的一些特征词,导致标引不充分,无法全面地反映整个文档的内容特征; ②无法从层次角度来检索,满足用户对非叶节点的检索要求; ③文档树中不同节点含有大量重复的词汇,直接进行索引导致占用大量计算机资源,检索效率也将大大受到影响。

3 一个新的索引体系

本文针对以上问题,给出了一个系统的索引方法: ①从文档树中的所有节点(包括叶节点和非叶节点)中抽取索引词条,抽取索引词的标准是根据该词汇的权重大小; ②对从不同节点中重复的索引词条进行相应的处理; ③建立倒排文档。这样就解决了标引不充分,无法进行层次检索以及占用大量计算机资源的问题,最终实现内容与结构的双重检索。

3.1　节点中词汇的权重计算

我们知道一篇文献通常围绕某一主题进行论述, 根据语义, 可将一篇文献视为一概念树, 上层节点元素所覆盖的内容范围比下层要广一些, 只有合适层上的节点元素被检索出来时, 才真正称得上信息检索。为了得到任意层次上的元素, 叶节点和内点(如章节、根节点) 都需要进行索引。以下是一个 XML 文档:

```
<书>
<著者>王晓东</著者>
<题名>计算计算法设计与分析</题名>
<章>分治法的算法策略是将原问题分解为规模较小的相同子问题,这些子问题是原问题的较小模式,为使用递归技术提供了方便</章>
<章>动态规划的算法策略是将原问题分解成若干个子问题,求解子问题,然后从子问题的解得到原问题的解</章>
<章>贪心法的算法策略是把原问题分阶段来完成。在各个阶段,选择那些在某些意义下是局部最优的方案,期望各阶段的局部最优的选择带来整体最优。</章>
</书>
```

图 4(a) 是如图 2 所示处理过的 XML 词汇树, 我们发现, 很多词汇重复出现在不同节点中, 如果直接进行索引, 会占用大量的计算机资源。选取能反映文档内容特征的词汇进行索引的基本原则是某个词汇出现频率较高且最终出现在叶节点中。一般来说, 从底层的叶节点来看, 文档中的基本概念被传递给它, 所以可以选来作索引词。考虑到文档的结构层次, 如果某个节点中一个词汇又频繁出现在它的直接后继节点中, 也能较好地体现文档的内容特征, 可以作为该节点的索引词, 词汇在节点中的权重与该词汇在其直接后继节点中出现有关[9]。式(1) 充分考虑了这方面的因素, 给出了词汇 t_i 在任一非叶节点 E_j 中权重的计算方法

$$\mathrm{Weight}(t_i, E_j) = \ln(1 + tf(t_i, E_j)) \cdot I(t_i, E_j) \quad (1)$$

其中, $Tf(t_i, E_j)$ 表示词汇 t_i 在节点 E_j 中的词频, $I(t_i, E_j)$ 是个熵, 它反映了词汇 t_i 在节点 E_j 的直接后继节点中的分布情况。

$$I(t_i, E_j) = \frac{-\sum_{sub_k \in E_i} tf(t_i, sub_k) \cdot \ln \frac{tf(t_i, sub_k)}{tf(t_i, E_j)}}{-tf(t_i, E_j) \cdot \ln \frac{1}{N(sub)}} \quad (2)$$

其中, sub_k 表示节点 E_j 的第 k 个直接后继节点, $N(sub)$ 表示这些直接后继节点的个数。段落是文档树中的叶节点, 对于段落中词汇的权重采用传统的 TFIDF 的计算方法, 所以对于段落中的某个词汇的计算公式如下:

$$\mathrm{Weight}(t_i, P_j) = \ln(tf(t_i, P_j) \cdot \ln \frac{N}{n_i} \quad (3)$$

其中, $\mathrm{Weight}(t_i, P_j)$ 表示词汇 t_i 在段落 P_j 中的权重, $Tf(t_i, P_j)$ 表示词汇 t_i 在段落 P_j 的词频, N 为信息源中总的文档个数, n_i 为信息源中出现词汇 t_i 的文档个数。

由式(1) 和式(3) 计算出了任意节点中所包含词汇的权重, 并且权重值范围为(0, 1) 。

3.2　索引词的选择

经过图 2 处理而成的词汇树中存在这样的情况, 某个节点中的一个词汇又频繁出现在它的直接后继节点中, 或者同一词汇大量重复出现在同一层次的不同节点中, 为了避免词汇重复, 在汇总树中, 如果某词汇出现在某一元素中, 那么它就不应该出现在它的后代节点中, 以避免概念交迭。任何一个内节点与它的直接后继节点是有关的, 如果一个词汇的权重比较高, 可被选来作索引词。特别地, 如果某个层次上的某个节点中的某个词汇的权重超过一定的阈值, 我们认为它表示的概念范围要广, 应该将其层次升级, 从它所在的节点中删除, 提升到高一层上, 从而使它代表更一般的概念, 避免了冗余。这个处理过程从底层的节点开始, 逐层向上计算, 直到文档树中的所有节点都被处理, 具体算法步骤如下[9]:

(1) 对每个叶节点, 根据式(3) 计算所有索引词的权重。

(2) 对每个内节点元素 E_j, 使用式(1) 计算所包含词汇的权重。

(3) 对于词汇 t_i,

if $\mathrm{weight}(t_i, E_j) \geq \mathrm{average}(E_j) + \mathrm{std\_dev}(E_j)$

then t_i 被选作为节点 E_j 的索引词汇, 并且将其后代所有该词汇删除。

(4) 重复(2) 直到根节点 $\mathrm{average}(E_j)$ 表示节点 E_j 中所有词汇权重的平均值; $\mathrm{std\_dev}(E_j)$ 表示一个标准值。

经过上述过程的处理, 形成的汇总树节点中没有任何两个词汇是重复的, 如图 4(b) 所示 。

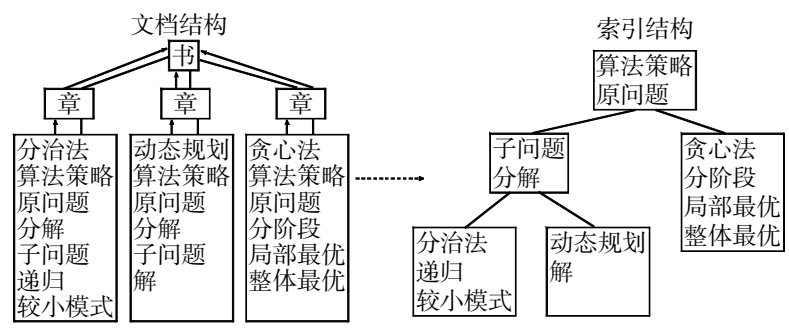

图 4　XML 的词汇树及汇总树

3.3　倒排文档的生成

对汇总树生成倒排文档(Inverting File) 结构如图 5 所示。

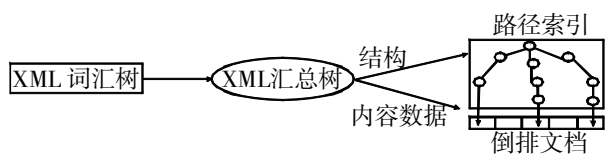

图 5　倒排文档结构

3.4　查询方法

查询可以用规则的表达式来说明查询要求, 如 query[(* section) with (“information”)] 表示查询包含词 “information”的所有节, 简单的方法是遍历倒排文档, 查找 “information”, 找到相应的在倒排文档中的 “information”索引项。它指出了文本中 “information”出现的位置, 然后搜索层次索引, 寻找包含该词汇的节或段落[1]。一般来说, 查询关键词附近的文本片断(一般是 50 ～200 个单词的范围) 是最好的查询结果[8]。

4　结束语

随着越来越多的信息以 XML 的形式表示和描述, XML 文档信息检索的研究也越来越受到重视。XML 的出现, 为信息的处理提供了内容与结构两方面强有力的支持。本文着重探讨了支持 XML 文档信息检索系统的索引技术, 给出了一个完整的索引方法, 包括计算 XML 文档树中的词汇权重, 从词汇树到汇总树的转换, 确定索引词条, 建立倒排文档。本文目前主要进行理论与算法上的研究, 在以后的工作中要逐步完善并实现这些算法。

(下转第 50 页)