

Web 日志挖掘中数据预处理方法的研究

李烈彪, 张海鹏, 周亚峰

(重庆大学 计算机学院, 重庆 400044)

摘要: Web 日志挖掘是目前网上智能信息检索和电子商务的主要研究课题之一。而数据预处理在 Web 日志挖掘中起着很重要的作用, 直接影响日志挖掘的质量和结果。介绍了 Web 日志挖掘数据预处理过程, 综述了国际上的研究现状, 及流行的处理方法。针对预处理步骤中的用户会话识别和路径填充进行了相应的改进。根据评估会话构造方法的标准, 通过实验对给出的新方法与其他方法进行了分析比较。

关键词: 数据挖掘; Web 日志挖掘; 数据预处理

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2007)07-0045-04

Data Preprocessing Method Research for Web Log Mining

LI Lie-biao, ZHANG Hai-peng, ZHOU Ya-feng

(College of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract: Web log mining is one of the main research domains in intelligent information retrieval system and electronic commerce. While data preprocessing has played an important part in Web log mining, directly influenced the quality of the Web log mining and its results. This paper introduces Web log mining data preprocessing process. Summarizes studies status and prevalent methods did in this area. Then improves the user session identification and path identification which are the processes of the data preprocessing. Finally according to measures for session construction methods, compares the performance of the new method to other session construction methods by means of experimental data.

Key words: data mining; Web log mining; data preprocessing

0 引言

自从 1991 年以来, WWW 已经发展成为拥有数亿用户、数十亿页面的巨大分布式信息空间, 而且其信息容量仍在飞速增长。但 Internet 是一个具有开放性、动态性和异构性的全球分布式网络, 信息资源分布很分散, 且没有统一的管理机构, 这就导致了信息获取的困难。绝大部分用户极容易在“黑暗”的网络中迷失方向, 也极容易在“跳跃式”访问中烦乱不已和在等待信息中失去耐心^[1]。

解决这些问题的一个有效途径就是 Web 挖掘。Web 挖掘就是把数据挖掘的技术应用到 Web 数据中以发现感兴趣的有用模式和隐含信息。Web 挖掘可以分为 3 类: Web 内容挖掘(Web content mining)、Web 结构挖掘(Web structure mining)和 Web 使用记录挖掘(Web usage mining)^[2]。其中 Web 使用挖掘就是通过挖掘 Web 日志记录来发现用户访问 Web 页面的模式,

从中可以提炼出设计者的领域知识、用户感兴趣程度、用户的访问习惯等, 进而得到优化站点结构、开展个性化服务以及用户访问控制等对站点设计者、经营者有用的决策性信息。文中主要讨论了 Web 日志挖掘的预处理的研究现状及其方法, 并在此基础上对用户会话构造和路径填充算法进行了相应的改进。

1 Web 服务器日志

因为 Web 服务器清晰地记录了网站访问者的浏览行为, 所以 Web 服务器日志(Web Server Log)可作为 Web 日志挖掘的重要数据来源。目前日志文件以多种数据格式存储在 Web 服务器上。最常见日志格式分为两种: 通用日志格式(Common Log Format)和扩展日志格式(Extended Log Format)。如下便是 Apache 服务器上两种日志格式的一般记录:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 - 0700] "GET/apache_pb.gif HTTP/1.0" 200 2326
```

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 - 0700] "GET/apache_pb.gif HTTP/1.0" 200 2326
```

收稿日期: 2006-09-20

作者简介: 李烈彪(1948-), 男, 重庆人, 副教授, 硕士生导师, 研究方向为建筑智能化、计算机控制。

“http://www.example.com/start.html” “Mozilla/4.08 [en] (Win98;I;Nav)”

其两者的主要区别在于:扩展日志中加入了引用项和客户端浏览器信息。

2 Web 日志数据预处理的流程及方法

2.1 数据预处理过程

数据预处理是为了将日志文件转换成数据库文件而进行的工作,其目的就是把 Web 日志数据转换为适合进行数据挖掘的精确数据。结合数据挖掘中遇到的问题可以把预处理过程分为如下几个步骤(如图 1 所示)。

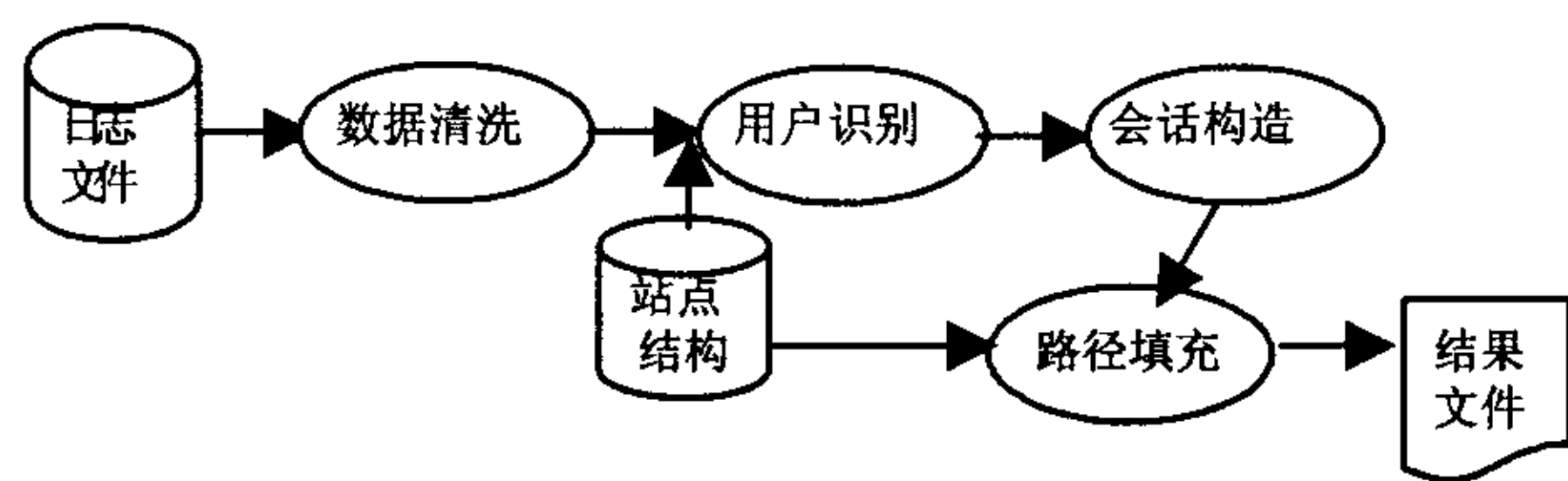


图 1 Web 日志挖掘数据预处理过程

2.2 数据清洗

数据清洗是整个数据挖掘工作的基础,在任何形式的 Web 日志分析过程中,清除服务器日志中不相关数据都是非常重要的。只有当服务器日志中表示的数据能够准确地反映用户访问 Web 站点的情况时,经过挖掘得到的知识才是真正有用的。数据清洗包括删除一些对于分析没有意义的数据,去掉访问中出错的纪录、用户请求方法中不是 Get 的纪录,及一些代理访问或系统产生的非人类请求纪录。然后将处理后的数据导入到关系数据库中,在进行进一步的知识发现^[3]。

2.3 用户识别

用户识别就是识别哪些用户访问了站点。通常在进行用户识别的过程中会遇到下面的典型问题^[3]:

● 单个 IP 对多个服务器用户访问会话:ISP 利用 Proxy 代理为用户提供服务,统一 IP 访问一个 Web 站点时可能是不同的用户。

● 多个 IP 对单个服务器用户会话:有些 ISP 对来自同一个用户的请求,会随机分配若干个 IP 中的一个给用户,这样一个用户进程会有不同的 IP。

● 多个 IP 对单个用户:从不同机器上访问 Web 的同一个用户因为不同的进程而拥有不同的 IP,这也使得追踪同一个用户变得复杂。

● 多个服务器进程对单个用户:这种情况发生在用户打开多个浏览器窗口,同时对同一个站点的不同部分进行访问。

● 单客户端对多用户:如家庭,很多人共用一台机器。考虑到这些问题并结合前人的方法,可以用下

列启发式规则来识别用户:

- 不同的 IP 代表不同的用户。
- 如果 IP 相同,则不同的操作系统或浏览器代表不同的用户。
- 如果 IP 和操作系统和浏览器都相同则根据引用页进行区别,如果引用页为空的话,则认为是一个新的用户;若引用页不空且多个用户包含此页,则把它划为时间上最近访问用户。

需要说明的是:使用上述方法并不能完全识别出用户,如用户在浏览用户一个网页时,直接在地址栏里输入 Url 而转到其他网页就会被认为不同的用户。

2.4 会话识别

会话(Session)是指用户在一次访问网站期间从进入网站到离开网站所进行的一系列活动^[3]。在跨度时间段较大的 Web 服务器日志中,用户可能多次访问了该站点,会话识别的任务就是把属于同一用户的同一次访问请求识别出来。

2.4.1 会话识别的方法

目前,会话的构造主要是基于启发式的方法:如基于时间的,依据站点结构的,给予引用的。现人们常用的算法有如下 4 种:

● H_{visit} :给用户在整个站点的停留时间一个上届,如果超过这个域值 θ 则认为新的会话开始^[4,5]。设 t_0 为会话初始页的时间戳,同一用户的一个 URL 请求的时间 t 如果满足 $t - t_0 \leq \theta$,则被加入当前会话,第一个满足 $t_0 + \theta < t$ 的页面成为下一个会话的初始页。一般 θ 取 30min。

● H_{page} :给用户一个页面停留时间域值 Δt ^[6],如果 2 个连续请求的时间间隔没有超过这个值 Δt ,这属于同一会话,否则分属于两个会话。 Δt 一般取 10 min。

● H_{Ref} :利用用户访问历史和参引页来划分^[5],如果一个用户的请求不能通过参引页上的链接进入,则很可能属于另一个会话。即当前请求的参引页没有在前面访问过的页面中出现,则是一个新的会话开始。

● MF(maximal forward references):最大向前参引模型^[7],即在一个用户会话里不会出现用户先前已经访问过的页面。如果用户在向前浏览到一个网页时,按下了“返回”按钮,则表示当前会话结束,一个新的会话开始。

2.4.2 基于时间和引用的启发式方法

该文给出的会话构造方法是结合上面基于时间、引用和页面停留时间三者的特点来对用户会话进行构造。基于这个出发点主要是考虑两个方面的因素:

(1)基于时间的方法在以往的应用中性能表现很

好;

(2)考虑该文主要是针对发现用户访问模式的应用,引入用户的浏览特性将有利于下步用户访问模式算法的实施。

算法主要思想如下:首先利用会话的时间特性来进行区分,这里取时间阈值 θ 为 30min。如果大于这个值则认为一个新的会话开始了,否则再根据用户的浏览特性和页面之间的连接结构确定最终的会话集:假设同一用户依次发出相邻的两个请求 p 和 q (其中 p 属于会话 S), t_p 和 t_q 分别表示页面请求 p 和 q 的时间戳($t_p < t_q$)。如果请求 q 的引用页面($q.refer$)曾经在会话 S 中出现过,那么 q 就属于会话 S ;或者 q 的引用页为空且 $t_q - t_p \leq \Delta$ (Δ 为时间延迟,通常小于 1min),那么页面请求 q 也属于会话 S 。

在给出算法之前,首先给出服务器日志和会话的形式化描述:

定义1 Web 服务器日志可以看成是按照时间戳排序的集合 $L = \{l_1, l_2, \dots, l_i, \dots, l_n\} (1 \leq i \leq n)$ 。 $|L| = n$ 表示日志集合 L 包含的元素数目,经过用户识别之后日志的纪录 $l_i = \{\text{userid}, \text{url}, \text{time}, \text{refer}\}$ 。此处的 userid 是经过识别用户后赋予每个用户的唯一识别码,下文提到的 userid 均表示此意。

定义2 当且仅当三元组 $r = \{\text{userid}, s_m, (< l_i.\text{url}, l_i.\text{time}, l_i.\text{refer} >, \dots, < l_i.\text{url}, l_i.\text{time}, l_i.\text{refer} >)\} (1 \leq i < j, l_i, l_j \in L)$ 满足 $l_i.\text{userid} = l_j.\text{userid}, l_i.\text{time} < l_j.\text{time}$ 时, r 被称为会话。 s_m 为给过构造后的会话标示。 $\text{length}(r)$ 表示会话 r 的长度,即会话 r 所包含的页面请求数目。

算法:生成会话集合 $\text{Generate_Session}(L, \text{SessionSet})$;

输入:经过用户识别后的 Web 服务器日志 $L, \theta = 30\text{min}, \Delta = 1\text{min}$;

输出:会话集合 SessionSet ;

$\text{Generate_Session}(L, \text{SessionSet})$

$\{ i := 1, m := 1;$

$\text{while}(i \leq |L| - 1)$

$\{ r := \{ l_i.\text{userid}, s_m, (< l_i.\text{url}, l_i.\text{time}, l_i.\text{refer} >)\};$

$j := i;$

$\text{while}(l_{i+1}.\text{userid} = l_i.\text{userid})$

$\{ \text{if}(l_{i+1}.\text{time} - l_j.\text{time} \leq \theta)$

$\{ \text{if}(l_{i+1}.\text{refer}$ 与任一 $l_k.\text{url} (j \leq k \leq i)$ 相同)

$\text{or}(l_{i+1}.\text{refer} = \text{null} \text{ and } l_{i+1}.\text{time} - l_i.\text{time} \leq \Delta$

$\})$

$r := r \cup < l_{i+1}.\text{url}, l_{i+1}.\text{time}, l_{i+1}.\text{refer} >;$

else

$\{ \text{add } r \text{ to SessionSet};$

$m := m + 1;$

$r := \{ l_{i+1}.\text{userid}, s_m, (< l_{i+1}.\text{url}, l_{i+1}.\text{time}, l_{i+1}.\text{refer} >)\};$

$j := i + 1;$

$\}$

$\}$

else

$\{ \text{add } r \text{ to SessionSet};$

$m := m + 1;$

$r := \{ l_{i+1}.\text{userid}, s_m, (< l_{i+1}.\text{url}, l_{i+1}.\text{time}, l_{i+1}.\text{refer} >)\};$

$j := j + 1;$

$\}$

$i := i + 1;$

$\} // \text{while}$

$\text{add } r \text{ to SessionSet};$

$i := i + 1;$

$\} // \text{while}$

$\}$

2.5 路径填充

路径填充的目的是为了补全访问日志中没有纪录的用户请求,获得用户的完整的访问路径,这样才能更准确地发现用户的访问模式。通常,用户在浏览网页时,由于本地缓存和代理服务器缓存的存在,使得用户通过按下浏览器上的“后退”按钮而得到的页面,而在服务器日志文件中却没有纪录。比如:访问 ABAC,由于缓存的存在,只能记录到 ABC,需要使用路径填充算法来进行补充。目前大多数路径填充算法都使用网站的拓扑结构来进行填充,方法的优点是准确率比较高,但是随着网站的信息越来越多,网站的拓扑结构会越来越大,这样就会很费时。文中利用日志文件中的引用记录和用户的历史访问记录对用户访问路径进行填充。该方法的思想是:假如 p 和 q 是同一用户的两个连续请求, $q.refer \neq p.url$ ($q.refer \neq \text{null}$),则页面 q 不是重页面 p 直接到达,这时在当前用户会话中查找等于 $p.refer$ 和 $q.refer$ 的 url 纪录,假如找到的纪录分别为 $s.url$ 和 $t.url$,如果两者相等,则直接插入,否则,把 $t.url$ 到 $s.url$ 的记录插入到 p 和 q 之间。如果在当前会话里面找不到,则在当前用户的其他会话里面查找;如果存在多个会话包含记录 $s.url$ 和 $t.url$,则取时间和 q 最近的会话进行填充。例如:一个用户的访问记录为 ABDECF,其对应的拓扑结构图如图 2 所示。

由于 D 到 E 之间没有直接连接存在,所以用户在访问 D 后利用了缓存而到达了 E,而 D 和 E 的参引页面都是 B,所以应在 D 和 E 之间添加页面 B;同样, E 和 C 之间也没有直接连接存在, E 和 C 的参引页面分别为 B 和 A,则需要把 B 到 A 之间的页面记录填入到 E 和 C 之间。填充后的完整访问路径则为: ABD-BEBACF。

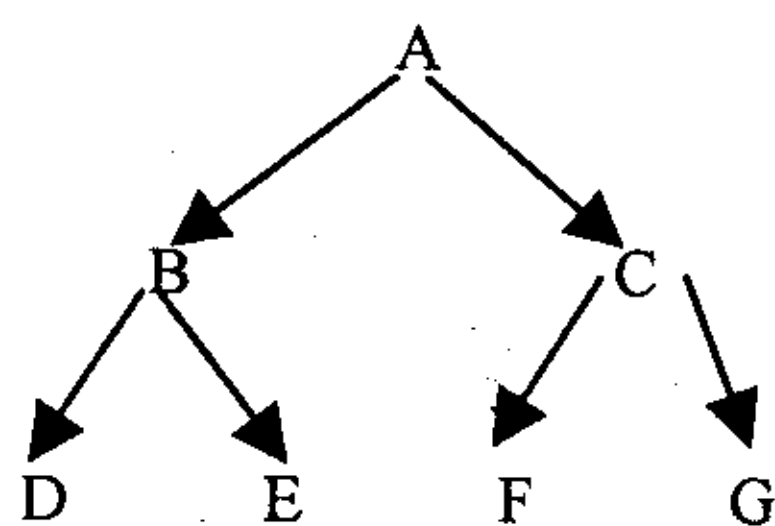


图 2 拓扑结构图

3 结果评价

不同的知识发现对复原会话的结果有不同的要求,有的关注完整性,有的关注连续性。例如:一个关注访问者购买倾向的网站,对于访问的网页的集合感兴趣,此时会话划分的完整性更重要;而一个希望改进自己站点结构的网站,更关心用户的访问序列,是否需要改变的链接,此时会话的连续性更重要。因此需要对算法的品质进行评价,为具体应用选择最适用的划分算法。最理想的重建结果就是得到的会话和真实会话完全相同。

目前常用的评价标准框架有两种^[4]:

1)一种评价标准是会话被算法 h 完整重建的程度。一般使用精确度和查全度这两个指标来衡量重建的性能。

精确度是用完全构建的会话数目与构造生成的总会话 R_h 数目的比值表示: $\text{precision}(h) = \frac{|R_h \cap R|}{|R_h|}$

查全度是完全构建会话数目与真正的会话 R 数目的比值表示: $\text{recall}(h) = \frac{|R_h \cap R|}{|R|}$

2)另一种评价标准是根据构造出的会话与真实会话之间的重合度来衡量重建性能的方法。重合度是真实会话 $r \in R$, 和重建会话 $c \in R_h$ 的最大重合序列值与真实会话序列值的比, 表示为: $\text{deg}(r, c) = \frac{|r \cap c|}{|r|}$ 。由于一个真实的会话 r 可能会被某一个重建方法分割成多个构造会话 c , 因此, 一个真实的会话的重合度是此真实会话与每个构造会话的单一会话重合度的聚集。一般, 这个聚集程度可以用单一会话重合度为变量的函数 f 表示, 用单一会话重合度的最大函数 $f(r, c, \text{deg}) = \max\{\text{deg}(r, c_h)\}$, ($c_h \in R_h$) 表示真实会话 r 的重合度。最后, 在评估构造会话方法 h 性能的时候, 可以用真实会话重合度的平均值函数 $g(h, \text{deg}) = \text{avg}_r\{f(r, c, \text{deg})\}$ 作为构造会话方法 h 的平分函数。

4 实验结果

在实验中采用的是重庆大学网站 2006 年 5 月 20 日的 Web 服务器日志数据。经过清洗、用户识别后得

到数据 6820 条纪录。本实验实现了文中提到的基于时间的方法 M_1 、基于引用的方法 M_2 和该文的方法 M_3 。实验数据如表 1 所示。在对上述 3 种方法进行比较时, 采用以会话时间的方法为基准, 用第一种评价方法来比较。通过数据可以得出: 方法 M_3 在查全度方面的性能表现比基于引用的方法 M_2 稍好, 两者在精确度方面的性能表现差不多。虽然方法 M_3 在查全度方面比 M_2 只高了几个百分点, 但是考虑到用户的会话数目比较大, 因此在绝对数目方面文中的方法性能还是有一定的提高的。

表 1 比较结果

方法	会话数	会话交集	精确度	查全度
M_1	1728	$ M_1 \cap M_1 = 1728$	$\frac{ M_1 \cap M_1 }{ M_1 } = 100\%$	$\frac{ M_1 \cap M_1 }{ M_1 } = 100\%$
M_2	2460	$ M_2 \cap M_1 = 1311$	$\frac{ M_1 \cap M_2 }{ M_2 } = 53.293\%$	$\frac{ M_1 \cap M_2 }{ M_1 } = 73.569\%$
M_3	2573	$ M_3 \cap M_1 = 1358$	$\frac{ M_1 \cap M_3 }{ M_3 } = 52.779\%$	$\frac{ M_1 \cap M_3 }{ M_1 } = 78.588\%$

5 总 结

通过对数据预处理的探讨研究, 不难看出 Web 日志数据预处理是 Web 日志挖掘的一个重要前提和基础。它为下一步的模式发现和模式分析打下良好的数据基础。介绍了 Web 日志挖掘的前期工作——数据预处理的过程和预处理的方法。并对预处理过程中的会话构造和路径填充方法进行了改进。下一步的研究重点是进一步改善数据预处理方法的总体性能, 期待寻找更准确、更有效、更合理的预处理方法。

参考文献:

- [1] Han Jia-Wei, Meng Xiao-Feng, Ang Jing. Research on Web Mining[J]. Journal of Computer Research & Development, 2001, 38(4): 405-414.
- [2] Han Jiawei, Kamber M. 数据挖掘概念与技术[M]. 范明等译. 北京: 机械工业出版社, 2001: 290-291.
- [3] 毛国君, 段立娟, 王实, 等. 数据挖掘原理与算法[M]. 北京: 清华大学出版社, 2005.
- [4] Fu Y, Sandhu K, Shih M. A generalization - Based Approach to Clustering of Web Usage Session[C] // Proc 1999 KDD Workshop Web Mining, LNCS 1863. [s.l.]: Springer - Verlag, 2000: 21-28.
- [5] Cooley R, Mobasher B, Srivastava J. Data Preparation for Mining World Wide Web Browsing Patterns[J]. Knowledge and Information system, 1999, 1(1): 5-32.
- [6] Spiliopoulou M, Mobasher B, Berendt B, et al. A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis[J]. INFORMS Journal of Computing, 2003,

(下转第 52 页)

于多个样本页面集,也要依次获得信息块的位置信息,所得到的训练集合即 $IBPATH_i (i \in \{1, 2, \dots, m\})$ 。

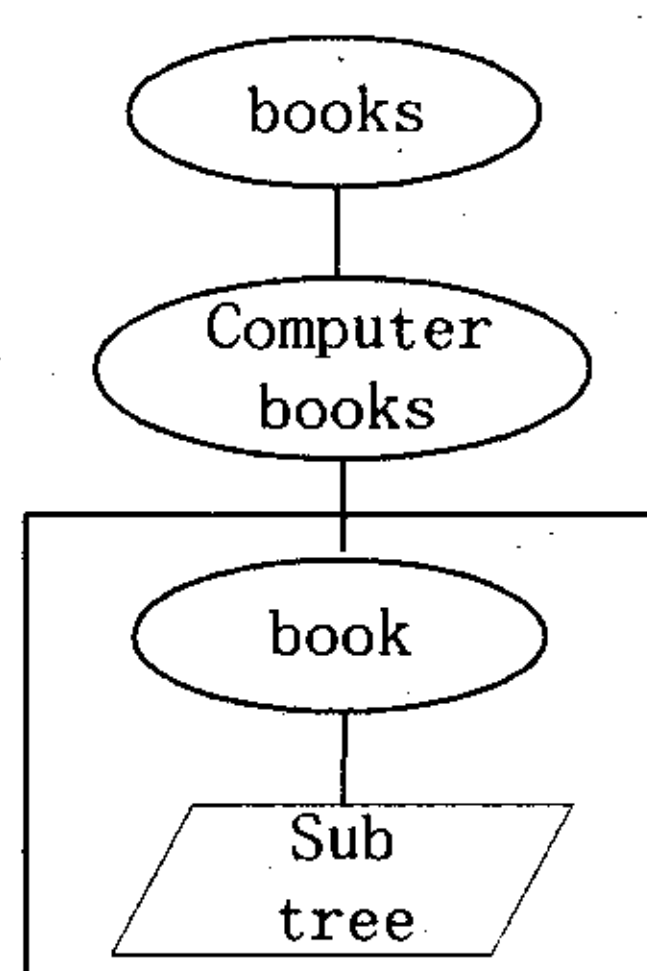
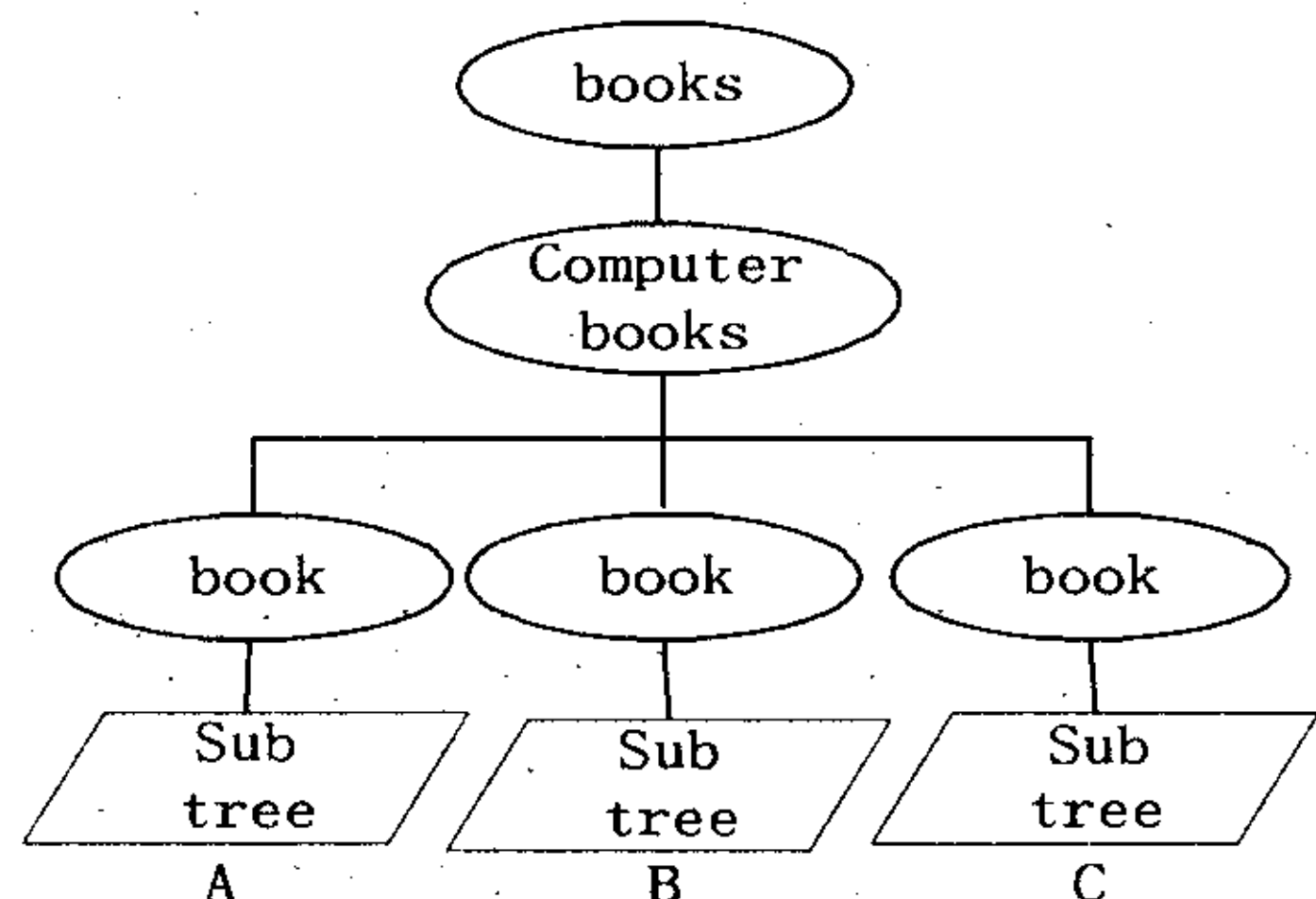


图 5 树模式 1 和树模式 2

在得到了待抽取信息块的路径集合之后,信息抽取实际上就变成了对信息块内部信息条集合的抽取。结合信息块以及块内信息条的定位信息,接下来仅需要将上述定位信息合并,根据每个结点的 Xpath 形成 XSLT 文件。该 XSLT 文件就是抽取规则。

当得出了抽取规则 XSLT 文档后,要构造一个进行信息抽取的 wrapper 仅需要执行这个 XSLT,便可得到 Web 抽取最终的结果。

2.4 多层向量空间模型

针对最终抽取出来的 XML 文档,文中在最后提出一种基于 XML 的多层向量空间的模型。在 XML 树型文档中,同一个词出现的位置可能有多种情况,即在每一层的结构中都可能有一个相同的词出现。为了更好地呈现各个层次的重要程度和表达的能力,便有了这里提出的基于 XML 的 N 层向量空间模型,在此也只做简单介绍。

N 层向量空间模型原本是利用 HTML 文件的特殊结构而改进的一种算法^[1]。由于一个文档具有不同作用的文本部分,如文档中的标题、摘要、正文、备注甚至标签等,尤其对标准的 XML 结构来说,这种层次感更强,它严格遵守了 DTD 的树形结构。在信息抽取以及查询匹配过程中,同一个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 4 个文档 d_1, d_2, d_3, d_4 ,这 4 个文件中都包含特征项 t ,并且 t 在这 4 个文档中出现次数都是 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 和 d_4 中, t 是被包含在文档正文中。运用传统的信息搜索引擎则会认为特征项 t 表达这 4 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。而对于文档 d_3 和 d_4 来说,如果 d_3 的正文长度小于 d_4 的文档长度,则相比之下也可以认为 t 对于文档 d_3 而言,它所代表文档内容的能力要比 d_4 强。因此,在对特征项 t 的检索过程中,相应文本的长度大小由小到大为 d_1, d_2, d_3, d_4 ,而相应的匹配值由大到小应

为 d_1, d_2, d_3, d_4 。

在 XML 结构中,可以简单地认为,越是接近于根的结点出现的特征词它所表达的能力就越强,或者可以自己定义出现在任意结点中的特征词作用是最大的。不仅如此,XML 的优势在于 Web 网页的隐藏数据都能通过 XML 的提取也加入到此模型中来,使其具有更好的表达能力。通过实验证明,这种模型下的信息检索比起传统向量空间技术速度更快、准确率更高。

3 结束语

文中对 HTML 网页使用了 XML 信息抽取技术,并对 XML 抽取结果采用了 N 层向量空间模型,总体上来说这种基于 XML 构架的向量空间技术在信息检索上能够发挥更好的功能。进一步工作将朝着网页上的语义块进行分析,从而不仅在结构上,也在语义上对网页有更深刻的研究和认识。

参考文献:

- [1] 王伟. 标记语言及 HTML 和 XML 的比较分析[J]. 现代图书情报技术, 2000(5): 22-24.
- [2] 刘斌, 陈桦. 向量空间模型信息检索技术讨论[J]. 情报杂志, 2006(7): 91-93.
- [3] 李萍. 浅析可扩展置标语言 XML[J]. 运城学院学报, 2005(5): 58-59.
- [4] 李剑波, 李小华, 董树明, 等. 一种基于 XML 的 Web 信息抽取方法[J]. 情报杂志, 2006(8): 49-51.
- [5] 周津, 朱明. 基于 XML 的网页信息自动抽取[J]. 计算机应用, 2004(S1): 225-227.
- [6] 王庆一. 多信息块 Web 页面中的抽取规则[J]. 计算机工程, 2003(9): 42-44.

(上接第 48 页)

15(2): 171-179.

[7] Chen M S, Park J S, Yu P S. Data Mining for Path Traversal

Patterns in a Web Environment[C] // Proc 16th Int'l Conf Distributed Computing System (ICDCS96). [s.l.]: IEEE CS Press, 1996: 385-392.