智能 Web 工具的体系结构*)

A Intelligent Web Tool Architecture

何炎祥 黄 浩 石 莉 张 戈 李 超

(武汉大学计算机科学系 软件工程国家重点实验室 武汉430072)

Abstract In this paper, we suggest the requirements of an intelligent Web tool and propose a system architecture. In particular, we design a learning agent and the underlying algorithms for the discovery of areas of interest from the user access logs.

Keywords User access pattern, Intelligent agent, Clustering

1 前言

Internet 正在迅猛发展,每天不仅有大量新的信息进入其中,而且有大量旧的信息经常被更新,有一些还以非常高的频率被更新。对个人来说,不可能得到所有这些信息和对这些信息做出的更新,这样就产生了帮助用户来获取、定位和管理 Web 文档的软件工具,称为 Web 工具[1]。

2 Web 工具的分类

根据 Web 工具的智能,Web 工具可以分为五个层次:

第0层 Web 工具,如 Mosaic,IE 和 Netscape,根据用户的指令直接获取文档,用户不得不通过文档的 URL 传给 Web 工具两个参数:文档存放在什么地方和如何获取这些文档。

第1层 Web 工具提供一个用户初始化的搜索机制以找到相关的网页。Internet 搜索引擎,如 Alta Vista(www.altavista.com)就是此类 Web 工具。大多数搜索引擎通过访问一个网页的索引进行搜索,当然,这个索引是非常巨大的。为了找到一个特定主题的相关网页,用户将描述这个主题的关键字提供给搜索引擎,根据文档与关键字的相似程度,将最后搜索到的文档进行排序后的输出。

第2层 Web 工具,如 WebWatcher[2]和 SIFT[3],对用户信息进行保存,并且有一活动的部件,只要找到新的相关信息它就会自动通知用户。SIFT 自动搜索新的网络新闻文章,找到用户感兴趣的新闻,然后通过 e-mail 将这些文章发送给用户。

第3层 Web 工具有一个用户信息的自学习和推理部件。用户信息包括用户感兴趣的主题和浏览方式等。Diffagent[4]和 Letizia[5]是两个实验系统,它们通过跟踪用户的浏览方式推断用户感兴趣的主题。

第4层 Web 工具能够学习用户和信息源两方面行为。为了更好地把最新的相关信息通知给用户,一个智能 Web 工具也需要了解信息源的行为。

本文中,我们主要对第4层 Web 工具进行讨论,提出一个实验系统的体系结构。

3 一个智能 Web 工具和体系结构

Web 工具系统有如下需求:

(1)系统应该能够自动发现用户感兴趣的主题,系统也应该能够了解到一段时间后用户的兴趣所发生的改变。在系统浏览 Internet 为用户搜索相关信息时,这些信息是非常有用的。

(2)系统应该能够学习用户的访问模式和信息源的更新模式。当信息源的文件更新时,它应该了解到,而且能够在用户发出请求之前就取回最新的版本。

(3)系统应该有效地利用网络资源。如应尽量避免会产生过分拥塞的搜索;多个用户的搜索应尽可能地集中,相似的搜索任务可以集合在一起。

(4)系统应该维护一个数据库和一个取回过的文档的索引,数据开采技术[6]可以应用到保存的文档上,以开采出不同的访问模式。

(5)系统应该兼容大多数 WWW 浏览器。无需其他软件,用户就可以使用标准 HTTP 协议和系统进行交互。

我们提出一个 Web 工具系统体系结构,见图1。

*)湖北省自然科学基金项目(98J075)和湖北省重点科技计划项目(982P0107)资助。何炎祥 教授,博士生导师,系主任,现从事分布并行处理和知识信息处理的研究。黄浩等 硕士研究生,研究方向为分布并处理和知识信息处理。

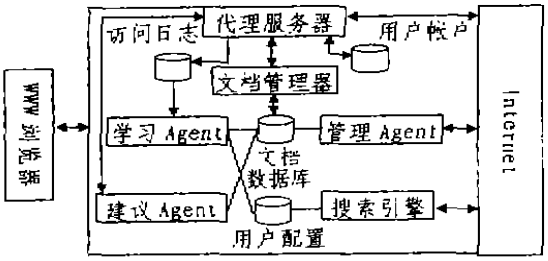

图1 Web 工具系统体系结构

WWW 浏览器 用户通过一个 Web 浏览器访问 Internet。在本系统中，用户只需将浏览器与系统的代理服务器(Proxy Server)相连。在一次会话中，所有的 HTTP 请求都通过代理发出，这样，系统可以得到一个用户访问日志。

代理(Proxy) 用户通过一个 WWW 代理服务器与系统进行通信。当用户发出一个 HTTP 请求时，请求被传送给代理，代理负责取回用户所需的 Web 文档，这些文档通过文档管理器被存放在文档数据库中。当代理收到一个 HTTP 请求时，它首先通过文档管理器进行检查，以判断所需的文档是否已经存放在文档数据库中。如果是，则将本地的备份传给用户；否则，需要访问 Internet。其次，代理需要知道每个用户所阅读的每篇文档，生成用户信息的日志。在本系统中，每个用户请求生成一个日志记录，其中包括用户 ID，请求 URL，访问时间和取回的文档。学习 Agent 使用日志中包含的信息来生成用户访问模式。我们将在第4节详细讨论学习 Agent 的工作过程。

文档管理器 是访问文档数据库的接口，负责存放和取回文档。这些文档既可以是用户所发出的 HTTP 请求所得到的，也可以是系统发出 HTTP 请求所得到的。文档数据库不仅提供文档的一致性存储，而且负责维护这些文档的全文字的索引。

学习 Agent 通过分析代理所生成的访问日志开采用户访问模式和感兴趣的主题，它为每个用户生成一个用户配置(user profile)。一个用户配置包含两类信息：a)用户感兴趣的主题。主题是用户感兴趣的关键字和词组的集合。例如，主题〈足球，0.7；甲 A，0.3〉表明总的来说用户以0.7的重视度对足球的信息感兴趣，尤其是甲 A(重视度为0.3)。这些关键字被用来驱动搜索引擎以开采信息；b)与时间相关的访问模式。有些文档经常被一个或多个用户访问，如报纸，股票价格，天气预报等。一个与时间相关的访问模式记录了文件位置和访问的周期。管理 Agent(Monitor Agent)使用这些信息来预取文档。

搜索引擎 在 Internet 上执行基于机器人的搜索。在搜索过程中遇到的感兴趣的文档就存放到文档数据库中，并建立索引。

管理 Agent 负责管理那些包含感兴趣信息的特定站点和网页。它有两个功能。首先，用户可以指明哪些文档应该保持最新。对于此类文档，管理 Agent 周期性访问它们，并学习信息源的更新模式，如一个网页以怎样的频率更新，何时更新，管理 Agent 使用这个知识，规划对网页未来的访问，保证文档数据库中它们始终最新。其次，规划那些用户经常访问的网页的预取。管理 Agent 从学习 Agent 生成的用户配置中得到这些知识。

建议 Agent 对新搜索到的文档进行排序，根据它们与用户配置的相似程度给出一个分数，它是一个由用户请求所搜索到的相关新网页所组成的列表，这个 Agent 也从用户反馈中进行学习，以改善未来建议的质量。

4 用户访问模式的开采

在这个智能 Web 工具中，学习 Agent 和管理 Agent 是两个重要的部件，在客户端，学习 Agent 负责从用户访问日志中识别访问模式，尤其是用户所感兴趣的主题；在服务器端，它又能够开采出网页的更新模式。管理 Agent 通过自适应地管理用户感兴趣网页的更新给客户提供服务。

用户访问日志所指出的文档包含一些未经处理的信息。为了开采出最感兴趣的主题，我们需要识别和提取最相关的关键字。本节中，我们主要讨论关键字的提取和描述中间使用的算法。对感兴趣主题的开采分为三个阶段，见图2。

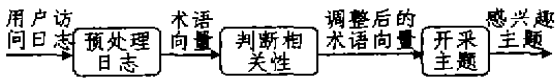

图2 从用户访问日志中开采感兴趣主题

第一步，学习 Agent 处理用户访问日志中记录的每个文本文档，生成一个术语向量(keyword，weight)。例如，一个术语向量(NBA，50)可以从一个体育文档中提取出来。

第二步，生成的术语向量可以直接用来开采用户感兴趣的主题。然而，在这些向量中存在大量的"噪音"。例如，一些网页可能不能提供任何信息给用户，却会被经常访问，因为它们有大量的超级链接。因此，学习 Agent 不得不判断每个文档的相关性，再根据文档的相关性调整从文档中提取的关键字的重视度。下面我们给出一些有用的方法。

(1)含有大量的 URL 的网页很可能是一个参考

链接的目录页。如果一个文档中包含的超级链接超过一个阈值，它可以被看成是一个参考页，可以向下调整它的相关性。

(2)用户花在一个文档上的时间可以从访问日志中计算出来。如果某个网页的访问时间很短，它很可能是一个参考页或是出于浏览目的的另一个"过路"页。因此，可以使用一个时间阈值来识别出这些网页，向下调整它们的相关性。

(3)多数情况下，在会话末尾访问的文档是处于对其内容感兴趣。为了识别出这些内容丰富的文档，可以从访问日志中构造一个浏览动作图。浏览动作图描述了被访问网页之间的前进和后退关系。访问日志中的每个文档构成浏览动作图上的一个结点；如果用户从一个文档移动到另一个文档，就构成浏览动作图上的一条边。在图中，可以标记出移动路径：从用户的初始化主页开始，结束于一个后退动作(一个后退动作是浏览器中的一个 go-back 操作)。每个移动路径可以看成是一个浏览会话，在会话末尾的文档有很大可能是内容丰富的文档。此外，加上时间因素，如果一个网页要成为一个内容丰富的网页，它不仅要接近一个会话末尾而且用户在它上停留了足够长的时间。一旦判定一个网页是内容丰富的网页，就向上调整它的相关性。

(4)浏览动作图中的结点可能有不同的输出。如果一个结点有许多输出，则表明这个结点的功能可能是一个参考页。因此，它的相关性应向下调整，它中间关键字的重视度也应减小。

(5)具有高相关性的文档中提取出的关键字需要根据它在文档中的功能做进一步的调整。例如，出现在标题中的关键字可以提高它们的相关性。

(6)高相关性的文档的 URL 可能含有反映用户兴趣的关键字。因此，从 URL 中提取的关键字应提高它们的重视度。

总的来说，第二步的任务是使用上述的方法来调整第一步所产生的术语向量的相关性。

第三步，从调整的术语向量中生成感兴趣的主题。我们使用聚类(clustering)技术来产生主题。作为输出，主题向量被截成一个预定义的长度，如(NBA，BASKETBALL，STADIUM，ARENA)。注意，主题向量代表一个用户感兴趣的一个领域。

本系统中，重分配方法被应用到调整后的术语向量集合上，将第二步中的相似向量进行聚类。两个术语向量的距离由它们的相似程度来衡量。两者越相似，它们之间的距离越短。两个术语向量 v_1 和 v_2 之间的相似性 $s(v_1,v_2)$ 为 v_1 和 v_2 的标准内积。当一个新的术语向量加入到一个聚类向量池中，它相对于聚类中心的距离可以计算出来，只要这个距离不超过一个特定的阈值，这个新向量将被最近的聚类吸收。一个聚类的中心是聚类中全部向量的平均值。如果由一个阈值所生成的聚类太多，则可以使用一个更小的相似性阈值或一个更大的距离阈值进行聚类，直到所生成的聚类足够小。

最后需要将聚类转换为主题。因为一个聚类中的术语向量和中心点十分接近，我们可以使用中心点代表所有的术语向量。然而，一个中心点可能有许多的关键字，这些关键字的重视度相对而言比较小，则需要对中心点的关键字做进一步的选择。通过预定义的长度阈值可以将中心点截短，或通过重视度阈值对中心点的关键字进行筛选。这样，系统的输出将是数量有限的术语向量。

结束语 这个 Web 工具的关键在于识别出文档的相关性和其中的关键字。以后的研究可以尝试增加学习 Agent 的学习能力，使其可以开采更多的知识[7~9]。

参考文献

1 Cheung D W, Kao B, Lee J. Discovering User Access Patterns on the World-Wide Web

2 Armstrong, et al. WebWatcher, A Learning Apprentice for the World Wide Web. Working Notes of the AAAI Spring Symp. Information Gathering from Heterogeneous, Distributed Environments. AAAI Press, 1995. 6～12

3 Yan T W, Garcia-Molina H SIFT-A Tool for Wide-Area Information Dissemination. In, Proc. of the 1995 USENIX Technical Conf. 1995. 177～186

4 Jones D H, Navin-Chandra D. Industry Net: A Model for *Commerce on the World Wide Web*. IEEE Expert, 1995 (Oct.): 54～59

5 Lieberman H, Letizia. An Agent that Assists Web Browsing In: Intl. Joint Conf. on Artificial Intelligence, 1995

6 Chen M S, Park J S, Yu P S Data Mining for Path Traversal Patterns in a Web Environment. In, R Cailliau, O. Nierstrasz, M. Ruggier eds. Proc. of the First Intl. World-Wide Web Conf. Geneva, 1994

7 Prodromidis A L, Chan P K, Stolfo S J. Meta-Learning in Distributed Data Mining Systems: Issues and Approaches. Advances in Distributed Data Mining. AAAI Press, 1999

8 Aronis J M, Kolluri V, Provost F J, et al. The world: Knowledge discovery from multiple distributed databases. [Technical Report ISL-96-6]. Department of Computer Science, University of Pittsburgh, 1996

9 Kargupta H, et al. Collective Data Mining From Distributed Vertically Partitioned Feature Space. In: Workshop on distributed data mining, Intl. Conf. on Knowledge Discovery and Data Mining New York, NY, USA, 1998