

基于.NET 的高校精品课程资源挖掘系统

王良文, 陈启安

(厦门大学计算机科学系 福建 厦门 361005)

摘要】 为提高教学质量, 建立规范性教学体系, 依托省级以上精品课程优质资源, 建立基于.NET 的高校精品课程资源挖掘系统。对于组成本系统框架的三个重要模块分别作了原理性说明, 同时说明了此系统的可行性并简单描述了系统的实现情况。

关键词】 搜索引擎, 资源挖掘, Web 服务

0. 引言

如何更好的完善高校教育体系, 如何采用多元化的手段补充现有教育模式的不足, 如何建立规范性教学体系是高校发展中永恒的主题。建设高校精品课程资源挖掘系统旨在提高教学质量, 建立规范性教学体系, 深度挖掘精品课程对教学的支撑作用, 最终达到加强学生的扩展学习能力、提高教师的教学技能、完善学校的教学系统的目的。通过科学的高校精品课程资源挖掘, 可使学校的教学管理部门建立更加科学、完善的课程体系; 使广大教师从中总结经验, 找出差距, 采取措施, 进一步完善教学工作; 使广大学生的知识面得到更宽广的延伸。

目前绝大部分高校都建设了不同级别的不同学科的精品课程系统, 但是这些精品课程的建设仅仅停留在完善学科建设的初级阶段, 而精品课程本身又是示范性课程中的精华, 所以, 如何更科学、合理的利用现有的精品课程体系, 进行深度的高校精品课程资源开发成了我们的研究重点。

1. 精品课程资源挖掘体系的建立

精品课程是具有一流教师队伍、一流教学内容、一流教学方法、一流教材、一流教学管理等特点的示范性课程。资源挖掘是从大量的资料数据中, 抽取潜在的、有价值的知识(模型或规则)的过程。精品课程资源体系是用 Web2.0 技术将现有的精品课程融会贯通成为一个精品课程资源平台, 给学校、教师和学生提供更好的学习发展的指导性平台; 精品课程资源挖掘系统是在该平台之上运用一系列科学的算法和智能化的资源挖掘手段对相关人群访问停留时间、检索情况、评价反馈等数据做出智能判断来更好的促进精品课程体系的建立、更好的促进高校教育体系的完善。优秀的精品课程资源是本系统建设的基石, 我们在设计精品课程资源挖掘系统的时候主要以省级和省级以上精品课程作为精品课程资源体系建设的基础; 先进的计算机信息网络技术是系统作用有效发挥的前提, 我们在实现精品课程资源挖掘系统的时候主要采用提供在线共享内容的 RSS(Really Simple Syndication) 技术和互联网智能数据挖掘技术。图 1 展示了精品课程挖掘系统的整个体系结构, 下面我们对该体系建立中的三个重要过程加以说明:

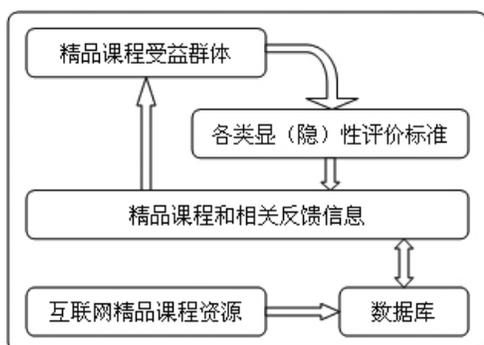


图 1 精品课程资源挖掘体系模型

1.1 精品课程采集

我们在采集精品课程资源的过程中重要采取两种方法: 一种是利用 RSS(也叫聚合内容, Really Simple Syndication) 技术, 通常直接通过一些精品课程汇总网站提供 RSS 输出来获取网站的最新更新内容; 另外一种方法是通过搜索引擎技术来实现的, 机制一般有两种, 一种是通过手工方式对网页进行索引, 它的缺点是 Web 的覆盖率比较低, 同时不能保证最新的信息, 第二种是对网页进行自动的索引, 在系统设计的过程中我们采用了此方法, 这种能实现自动的文档分类, 实际上采用了信息提取的技术。通过以上两种方法我们将采集到的信息放入数据库中慢慢组成系统的精品课程资源平台。

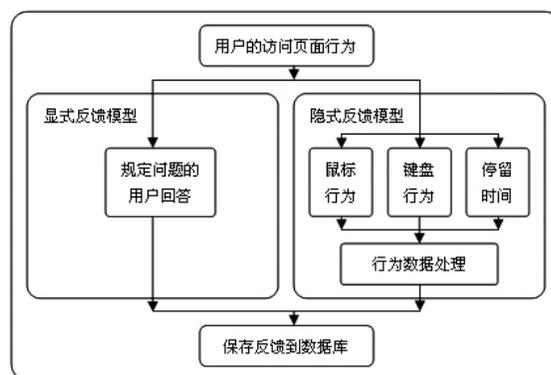


图 2 用户的显(隐)式反馈模型

1.2 课程信息聚合

课程信息聚合是指系统的智能主体在丰富的数据结构的基础上, 运用事先设定的思维逻辑对现有的课程资源进行重新组合。每一个精品课程都是一个完善的体系, 在众多精品课程之中用户如何获取其想得到的信息成为重点。课程信息聚合功能可以将各类课程的信息通过分组、分类、分级别等方式重新组合, 生成新的更有价值的信息资源提供给不同的用户群体。

1.3 用户的显(隐)式反馈

显示反馈是指用户在使用系统的过程中按照系统事先设定好的问题进行回答, 采用显示反馈的优势在于反馈结果清晰明了, 劣势是加大了用户操作的负荷以至于用户不愿提供相关反馈。隐式反馈是指不通过直接询问用户的方式, 由系统分析用户的页面停留时间、检索情况、访问历史等信息, 从而得到用户的隐式反馈, 隐式反馈不仅节约了用户的时间同时拓宽了反馈信息的途径。结合两者的优点, 系统可以得到更多更真实的反馈信息反过来提供给用户, 使用户得到更多的益处。

2. 系统的设计思想

2.1 搜索引擎的算法设计

搜索引擎的原理基本分为三步: 从互联网上抓取网页 建立索引数据库 在索引数据库中搜索排序。依照此原理我们在系统的 Web Service 中加入了 WebRobot 中间件, 实现方法如图 3:

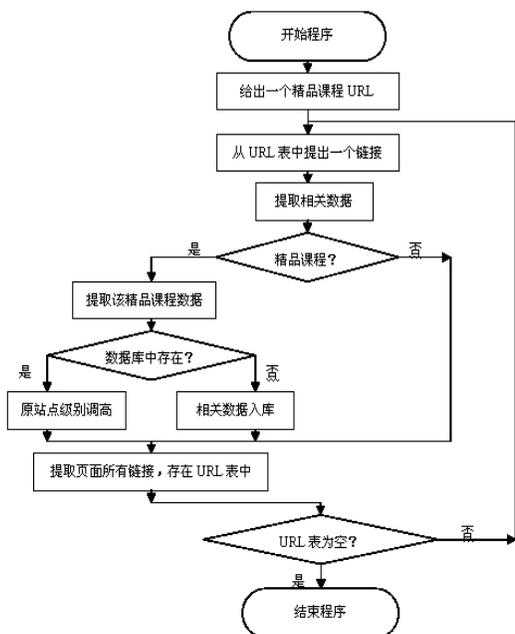


图3 搜索引擎的算法设计

2.2 信息聚合的原理

由于信息的不断增多会导致数据量的迅速增大, 为了为用户提供更快速的信息检索环境我们将所有信息进行整合, 每天凌晨三点是系统的不繁忙阶段, 此时由系统自动生成相关固定数据表达到方便用户检索的目的。根据系统的前期运行, 我们统计出最常见检索包括课程名称检索、课程负责人检索、课程类别检索, 根据实际情况我们设计如下: 课程名称检索关系模型: tabCourseSearch (CourseId, CourseName, CourseInfo)/课程检索表 (课程表编码, 课程名称, 课程简介), 用户在根据课程名称检索的时候, 我们将 tabCourseSearch 表的内容从数据库检索到用户本地机的一个数据存储对象 (DataSet) 里面, 然后让它给需要数据的数据窗口对象控件共享数据, 当用户确定需要完整数据的时候再访问数据库拿到全面数据, 这样不仅可以提高用户访问速度, 更可以减缓服务器压力。其它几种信息聚合方式与此雷同, 文中不再赘述。

2.3 用户的隐式反馈

我们在采集隐式反馈信息的过程中, 采用了页面内嵌 JavaScript 的方法来记录每个行为的分量值, 然后从每个分量中选择一个进行评价。设反馈分量集为 $T = \{T_1, T_2, \dots, T_n\}$, 其中 n 表示反馈分量个数。行为分量对应分数集为 $V = \{V_1, V_2, V_3, V_4\}$; $V_1 = 100\%$, $V_2 = 80\%$, $V_3 = 60\%$, $V_4 = 40\%$ 则某个用户对某门精品课程的一个分量指标的评价 $P_i = T_i \times V_i$, 其中 $i \in [1, N], j \in [1, 4]$ 。某个用户对某门精品课程的评价结果为:

$$\sum_{i=1}^N P_i = \sum_{i=1}^N T_i \times V_i$$

记为 K , 所有用户对某门精品课程的平均反馈结果就为

$$U = \frac{\sum_{i=1}^M K_i}{M}$$

其中 M 表示实施隐式反馈信息的人次。

3. 系统设计思路与实现

3.1 系统架构

精品课程资源挖掘系统采用了 ASP.NET 2.0+SQL Server 2005 的系统解决方案。在系统运行的过程中程序的运算速度、互联网的搜索速度和数据库的采集速度都得到了大幅度的提高, 满足了大用户群、大并发数、大数据量的需求。同时对系统进行了分层设计, 大量使用中间件, 并通过 web service 技术进行分

布式数据运算, 合理有效地降低了系统的负荷, 极大地提高了系统运行效率。整个系统框架如图 4 所示。

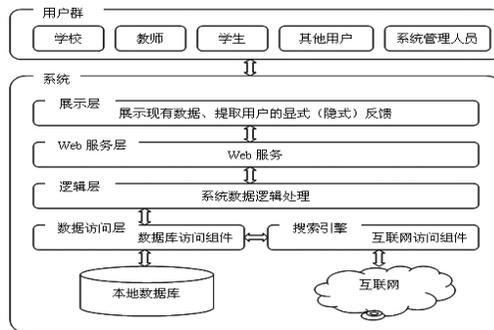


图4: 系统框架图

精品课程资源挖掘系统所需要的数据分为两部分, 一部分来自于本地数据库, 另外一部分来自互联网数据 (各个不同地域的国家级、省级精品课程等基础数据), 这些数据是利用 web service 从精品课程资源平台中获取的。

3.2 系统实现

目前精品课程资源挖掘系统仅在本校试运行, 系统实现使用 B/S 模式, 所以只要在本校局域网的任何一个终端都可以随时随地访问本系统, 通过本系统获得所需要的资讯。图 5 是精品课程资源挖掘系统的实现界面。

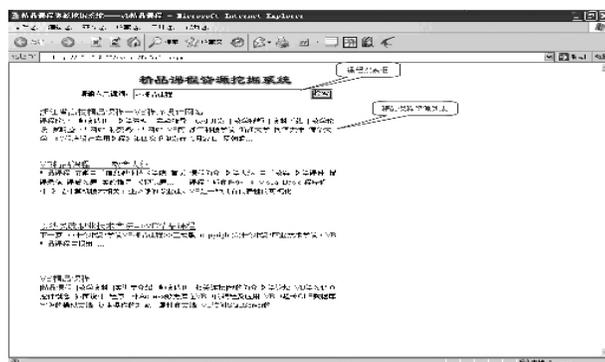


图5: 精品课程资源挖掘系统的实现

界面分为两大部分: 课程搜索部分和资源列表部分。实践证明: 精品课程资源挖掘系统达到了为我校的教学管理部门服务的目的; 使广大教师从中总结经验, 找出差距, 采取措施, 进一步完善教学工作; 使广大学生的知识面得到更宽广的延伸。

4. 结束语

经过一年来各个层次的使用群体的体验, 发现精品课程资源挖掘系统运行稳定, 数据处理速度快, 采集信息及时、准确, 对学校的学科建设、教师的教学水平、学生的知识宽度延伸有很大的帮助, 受到我校教学管理部门和广大师生的欢迎, 这说明本系统的结构设计和开发技术达到了较高水平, 具有很高的社会推广价值。

精品课程资源挖掘系统的实现思想还可以应用于其它互联网资源挖掘系统当中, 比如互联网新闻资源挖掘和商业资源信息挖掘等系统。当然, 实现思想还存在不足之处, 若各位同仁有更好的思想, 望共同学习、共同进步。

参考文献:

1. 脱建勇, 王嵩, 李秀, 刘文煌. 精品课共享中的推荐系统框架与实现[J]. 计算机工程与设计, 2006, 第 27 卷第 17 期: 3119- 3122
2. 何晓阳, 吴强, 吴治蓉. HITS 算法与 PageRank 算法比较分析[J]. 情报杂志, 2004, 第 23 卷第 2 期: 85- 86
3. 贾志娟, 胡明生. 基于网格的 Multi_Agent web 文本挖掘系统[J]. 微机计算机信息, 2006, (21): 266- 268