

信息检索

Internet网

Agent

网上信息

(23)

计算机科学2000 Vol. 27 No. 6

82-86

网上信息 Agent

The Information Agent on Internet

赵加欣 陈跃新

G354.4

TP18

(国防科技大学计算机学院 长沙 410073)

Abstract With the development of Internet, a lot of kinds of information agent applied for the net emerged, which have different functions and use different kind of learning algorithm. This paper discusses the structure and function of the Information Agent.

Keywords Learning, Information retrieval, Information filter

1 引言

随着网络的发展,网上信息资源越来越丰富^[1],这种网上信息具有分布、动态变化、结构复杂等特点,使得用户根本无法完全了解庞大的、瞬息万变的信息资源。此外,用户还要处理来自网上的大量信息(如电子邮件),即使是无用信息也要经过阅读后才能决定是否删除,大大影响了工作效率。

在这种情况下,传统的功能固定的搜索引擎和邮件管理软件已经不能满足用户需求了,Agent 技术则越来越显示出其优势。Agent 的自主性和学习性使得它既可以在不受用户监控、指导的情况下由自己的决策机制决定采取何种行动,与用户并行工作,又可以学习记忆用户的兴趣爱好,建立用户模型来指导自己的决策,使之符合用户要求。Agent 的这两个特点使它非常适合于对网上分布式资源的管理工作。通过一段时间的学习,Agent 可以在一定程度上掌握用户的喜好,帮助用户在网上检索信息,或按用户的习惯处理一些日常事物,成为用户很好的助手。

2 信息 Agent

Agent 的分类方式很多^[2],应用于网上的 Agent 称为信息 Agent (Internet/Information Agent),主要进行网上信息资源管理,有时也将之归为界面 Agent,是联系用户与网上信息的中介。

较为成功的信息 Agent 的应用范例有网上信息检索、过滤系统(如 Musag 系统^[3], Amalthaea 系统^[4]),网上广告业务及 E-mail 服务(如 Magi 系统^[5]),电子商务(如 MAGICA 系统^[6])等。还有一种 Agent 可嵌入浏览器,在后台观察用户行为,抽取行为模式,作为进一步服务的依据。(当然,它具通用性,可

嵌入任何一种应用程序,在这里不做讨论,如 Open sesame! 系统^[7])。

信息 Agent 在开发中遇到的问题既有 Agent 系统共性的问题,又有因网络应用而带来的新问题。具体说,主要有 Agent 组织方式的选择、Agent 的学习能力及通过学习完善知识的适应能力。此外还有 Agent 之间的通信语言和交互内容(包括 Agent 与用户间的交互),每个 Agent 的结构、功能、用户模型的表示方法以及 Agent 系统的管理、维护方式,等等。

3 信息 Agent 的体系结构

依据组织方式,Agent 系统的体系结构分成单 Agent 系统和多 Agent 系统。多 Agent 系统又可用集中式和分布式两种方式实现,多 Agent 合作较为适应网上资源丰富、变化快、分布等特点。

3.1 单 Agent 系统体系结构

单 Agent 系统由一个 Agent 完成学习并完成用户任务的动作,一般分成前台和后台两个部分,主要处理用户事件(键盘、鼠标操作,访问站点频率等)如图1所示。Agent 在后台抽取用户行为及系统信息的特征形成知识库,然后在前台利用知识库来处理新信息,前台、后台并行工作,提高了效率。

单 Agent 系统的实现可采用混合式的和一般式的。一般式系统就是选择一种算法实现 Agent 的学习功能,结构基本与图1一样。混合式的系统由多个以不同体系实现的功能块构成。如 Open Sesame! 系统是应用一种混合神经网络/专家系统体系结构。该系统的学习引擎是一个支持连续学习的非受控神经网络;推理引擎则是一个小型的专家系统。

单 Agent 系统对用户事件的处理要通过用户确认,与用户交互可以使用主动式(即 Agent 中断用户

的方式)或被动式(即用户访问 Agent 记录信息的方式)。主动式交互中允许用户间接(如撤消或增加对某些事件的监测)或直接控制中断频率;被动式则完全由用户决定何时查看 Agent 的工作结果。另外,还可以提供专门的 GUI(图形用户界面),使得交互更直观、易于使用。

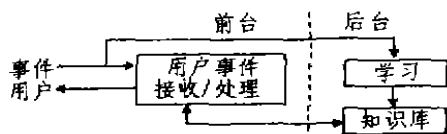


图1 单 Agent 系统体系结构框架图

单 Agent 系统的学习可使用增量式或更新式的。增量式学习保留知识库中内容,系统只在 Agent 学到了新知识时才扩充,其余时间是稳定的。更新式学习可在两种时机下更新知识库内容:一是利用时间戳,默认新近获得的知识更有效,定期删除旧的知识,从而使系统达到一种知识数量上的动态稳定;另外可采取延迟处理,在 Agent 工作较空闲时集中学习,系统在一段时期内是不变的。

3.2 多 Agent 系统体系结构

多 Agent 系统将任务细化,采用多个 Agent 分别完成。由于多个 Agent 间的交叉通信而形成一个网状结构,基本框架如图2所示,这里将就各 Agent 间的通信与合作方式,每个 Agent 的结构、功能、生存周期的定义以及各自保留的用户模型,系统稳定状态的到达等问题进行讨论。

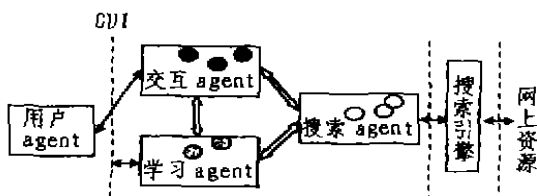


图2 多 Agent 系统体系结构框架图

3.2.1 集中式体系结构 集中式系统一般面向单个客户机,Agent 间可采取支持式合作和竞争式合作。不同的合作方式对学习算法的选择、学习结果的存储方式、最终用途,从而对 Agent 生存周期、系统稳定状态的定义都有直接影响。

支持式合作系统完成每项任务只需要一个 Agent,它们分工明确,互相服务又互相依赖,其间不存在资源竞争。比如一个由四种 Agent 构成的系统:学

习 Agent(LA),搜索 Agent(SA),交互 Agent(CA),结果 Agent(RA)。总任务是在网上检索并过滤用户所需的信息。Agent 间的支持合作体现为:SA 搜索结果先送给 LA 学习,再送给 RA 返回给用户,而 LA 学习的知识又是 SA 搜索及 RA 将结果排序的依据,CA 则是用户与其他 Agent 之间通信的桥梁,转换请求与应答的表达式。

竞争式合作系统定义多个 Agent 完成每项任务,完成同一任务的 Agent 以竞争方式来获得提供服务的权力,而完成不同任务的 Agent 之间还支持合作。经济模型是一种典型的管理 Agent 竞争的机制,仍以检索过滤网上信息的应用为例,考虑两类 Agent:IF(信息过滤器)进行信息过滤,当用户请求与它学到的知识相匹配时接收该请求;ID(信息发现)进行信息发现,利用搜索引擎为某个 IF 搜索所需的信息;同时引入 credit(自信度)变量,credit 变量类似商业活动中货币的职能,它还有决定 Agent 生命周期的作用。经济模型实现如下(见图3):

1) IF 接收用户查询请求后向所有的 ID 发出包含一个 credit 值的请求;

2) ID 按 credit 值并参考交易历史将接收的若干个 IF 请求排序,取 credit 最大的为之服务;

3) IF 接收若干个 ID 返回的信息,选择与自己的知识最相匹配的接受,并“付给”提供此信息的 ID 它曾许诺的 credit;

4) IF 将接受的信息返回给用户;用户给出的反馈信息将被转化成 credit“付给”提供信息的 IF;用户越满意,credit 值越大,否定时,为负值。

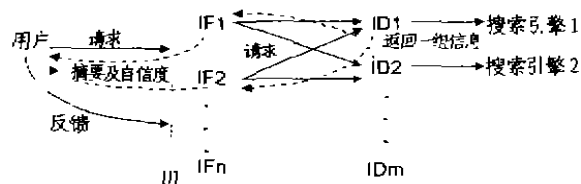


图3 ID、IF 竞争合作模型

无论是哪种合作方式,Agent 都要保留自己可识别的用户模型,而用户模型表示方法各不相同。

多 Agent 系统中,各 Agent 间的通信是一件重要的事情。简单的方法如“黑板”通信,只在“黑板”发生变化时 Agent 才访问它。另外还可以采用专门的 Agent 通信语言(如 KQML 语言)或 E-mail 作为通信手段,主要是告知其它 Agent 自己知道的用户信息,缩短学习周期。有些系统中 Agent 发出的请求和返回的信息都是其自身的一部分,这也是一种高效的通信方式。

Agent 与用户的交互通过用户界面实现: Agent 将找到的信息描述给用户, 用户给出反馈, 指导 Agent 改进工作。用户在这里也被视为一类 Agent 来建立模型, 协助软件 Agent 完成任务及完善服务。

不同的合作方式下 Agent 生存周期采用不同的定义。支持式合作的系统中完成每项任务的 Agent 只有一个, 一般不变化, 通过学习变化 Agent 保存的用户模型; 而竞争式合作系统中完成一个任务的 Agent 有多个, 则可采用“优胜劣汰”原则加以淘汰或再生, 破坏那些不适用于某个任务的 Agent, 但仍有同类的其它 Agent 在工作。如经济模型中若 Agent 总是提供用户不满意的文档, credit 就会不断减小, 小到一定程度时, Agent 就会被破坏而淘汰; 而 credit 本身也不断衰减, 使得 Agent 在总争取不到提供服务的机会时也会被淘汰。同时, 系统为达到平衡而用特定算法生成新的 Agent。由此可见, 支持式合作系统因所学用户模型的稳定而稳定, 竞争式合作的系统则因 Agent 群 (population) 的相对稳定而稳定 (即基本无 Agent 被破坏, 或被破坏与新生的频率较小)。后面将会看到, 采用不同的学习算法是两种系统在 Agent 生存周期与稳定性上存在差别的根本原因。

另外还有一种简单竞争式的系统实现方法。Agent 间的竞争体现所代表用户间的竞争, 系统用简单算法决定为哪个用户的 Agent 提供服务环境 (如“广告决策算法”^[3])。Agent 要能代表用户给出报价, 则在计算中就要考虑用户的兴趣所在, 因此同样有知识获取的问题。此种 Agent 不互相通信, 只要用户不退出, 其代理 Agent 便存在。

3.2.2 分布式体系结构 集中式 Agent 系统利用已有的信息检索、过滤软件, 将分布于网上各处的信息搜集来再作处理, 这种方式在以传输字节量计价的情况下是很昂贵的, 而传来的信息也并不都有用。因此, 出现了分布式的 Agent 系统来管理网上分布式资源。为实现分布式结构, 引入一个“地点” (location) 的概念, 即一个将工具、材料封装在一起, 使能并限制 Agent 运行的环境。其中“材料”是位于任意节点上具有不同结构与性质的资源, “工具”针对不同节点也具有不同功能特性。地点概念既易于 Agent 系统的分布实现, 又使得服务 Agent 结构更加简单, 它们只需携带用户信息和自己的功能描述在不同的地点间移动, 对于“地点”定义的语境的敏感性, 可使 Agent 到达不同“地点”时根据自己的功能利用该地点处的工具、材料为用户提供更加快速高效的服务。

分布式系统中各 Agent 的通信方式可采用与集中式系统相同的方式。作为用户代理的 Agent 一般是永久可重用的; 提供服务的 Agent 则可采用支持或竞

争式的合作模型。整个系统的稳定性是由分布于每个站点上的部分系统的稳定性决定的。

由于分布式系统中地点的信息资源被分布式存放, 因此要求 Agent 能很好地了解各种信息格式, 理解各种目录结构。当前, 只有以约定方式组织信息 (如 HTML 页) 才能较好地实现这一点, 但是显然在不同种目录信息间达成一致是很困难的。

4 学习算法

选择一种恰当的学习算法是 Agent 系统开发及应用的关键问题之一, 也是使得用户建立起对 Agent 的信任, 从而使 Agent 应用得到推广的关键。信息 Agent 网上服务大体分为两类: 一种是进行信息管理, 另一种是对用户进行跟踪。因此, 学习也可分为两类: 一种是信息特征抽取, 学习对象是文档, 采用记忆学习、关键字学习、学典学习等算法; 另一种是用户行为模式的抽取, 学习对象是用户操作行为, 采用模式学习算法。两者是从不同角度建立用户模型, 指导 Agent 完成任务。

4.1 信息特征的抽取方法

记忆学习 这种学习算法最简单, 一般只将从用户处得来的信息直接转化为内部表示, 存放在知识库中。这种学习方式还可以分为“学徒式”的和“专家式”的。前者是由 Agent 执行一定的算法, 在前台或后台将获取的信息转化为知识; 后者则由系统设计者通过与专家交谈, 获得问题域的知识, 然后将它们编辑在 Agent 的知识库中。这种划分是由系统功能决定的。面向单个用户的应用采用前一方法, 对每个用户渐渐适应; 面向多用户的应用采用后一种方法, 系统要选出一个用户为之服务, 这就必须事先掌握每个用户的一些知识, 使得开始时的选择基本为每个用户满意。记忆式学习虽简单但灵活性差, 需要足够大的存储空间。

关键字学习 又分两种情况, 一种是将搜索到文档中的字按出现频率、出现位置提取出来, 形成特征集; 另一种方法是将提取的关键字转换后存放起来。

对第一种方法, 我们以邮件 Agent Magi (一个处理用户电子邮件的系统) 为例介绍两种算法: CN2 和 IBPL1。这两种算法恰好采用了更新式学习的两种更新策略。

定义邮件的特征域为 FROM、SUBJECT、BODY。

CN2 是一种迭代式的算法, 用“分而治之”的方式归纳出一组规则集。首先将从收到的邮件中抽取各域的关键字而形成的特征集全排列, 使每个域只包含一个特征, 再为每个排列加上预测动作得到 CN2 的测试实例 (training example)。设定一些属性测试的 complex 集合, 通过用测试实例测试及拉普拉斯错误估算、

相似比统计方程^[3]两公式计算,反复选择当前最好集,迭代若干次后,将剩下的 complex 结合成规则。这样,抽取新邮件的特征集后,依据规则可预测采取的动作。当然可能一个邮件产生许多实例(全排列的结果),点火不同的规则而产生不同的动作,CN2为它们分别生成一个自信度,通过与阈值比较及相互比较,找到合适的那个提供给用户。

IBPL1 是对记忆算法的扩展,它处理的实例允许包含特征集,即每个域有多个特征值。实际上,IBPL1是直接为 CN2方法中生成的特征集定义一个类后存储起来,完成学习的,此后,通过距离公式^[4]计算出由测试实例得出的特征集与由新邮件得出的特征集之间的差距,就可决定新邮件属于哪一类。

IBPL1法不归纳规则,在新的训练实例生成时,就立刻被打上时间戳加入测试集中,旧实例会被去掉;CN2法则定时更新规则库(一般每天一次),生成的训练实例只能先放起来,这是它们最大的区别。虽然CN2法较复杂,但在许多方面性能比IBPL1法好。

对第二种方法,我们以进化生态系统 Amalthaea 的加权关键字向量生成器(WKVG)为例介绍,学习结果以向量存储。

Amalthaea 系统中负责学习的 Agent 由两部分组成:显型(phenotype)和基因型(genotype),其中显型是不可进化的部分,由基因型和处理它的可执行指令构成。基因型是可进化的部分,由加权关键字向量和“所过滤文档的作者”等其他必要信息构成。文档都由WKVG 转化为加权关键字向量,体现用户兴趣的文档(可从历史中获取)转化为向量后,即完成了学习。加权关键字向量即将文档中关键字过滤出并赋以权值存放在一起。每个文档对应一个向量,每个向量是一个 Agent 的组成部分。

Agent 以经济模型管理,向量的存储方式使得新 Agent 的生成方法很简单,只要交换两个 Agent 关键字向量(可进化的部分)的一部分就可产生两个新的 Agent(“两点杂交器”)。用算法来产生具有新知识的 Agent 有一定的随机性,但最终会由经济模型进行“自然选择”,“适者生存,不适者淘汰”。

字典学习 这是一种对相似概念的学习,可以解决当前文本搜索引擎的两大问题:一是用户询问中关键字模糊,致使检索出与论题无关的文档;二是相关的文档因不含询问中的特殊关键字而未被推荐给用户。我们以 Musag 系统的学习方法为例介绍。系统的学习 Agent 为用户提供的每个关键字维护一个字典。字典中每个字有两种权:一种是搜索权,只有文档中所含的字典中字的搜索权之和大于某一阈值,该文档才被学习;另一种是学习权,只有学习权大于某一阈值的字才

可存在于字典中,Agent 学习的不是与关键字含义相近的字或词,而是那些在关键字所在上下文出现频率高的字词,即所谓与关键字相关的概念。这种模糊概念学习的方法使得最后提供给用户的信息质量更高也更准确、全面。但这要等足够多的学习周期,短时间的学习不精确性是很大的,有时会与关键字原本含义有很大偏差。

最初的字典从用户给定的文档学来,一般 Agent 通过一段时间的学习及修改字典,最终可得到一个关键字的相对稳定的字典。比较 Amalthaea 系统的关键字学习,两者 Agent 生存周期的定义区别可以找到原因了,并且可以看出,Amalthaea 系统破坏一个 Agent 后,该 Agent 所有知识便消失了,而字典学习中字典的变化是循序渐进的。

4.2 用户行为模式的抽取方法

用户行为模式抽取即通过监视用户行为,分辨出重复模式,目的在于将之自动化处理或构造用户模型。所用算法的关键是识别重复模式的方式。

以 Open Sesame! 系统的学习引擎为例,它提供一种简单、结构化的语言来为第三方应用系统环境及用户事件建立域模型和事件模型。用户动作被描述为一个事件,并形成事件流输入到学习引擎进行模式学习,学习引擎利用域模型和事件模型中定义的相似性、可比性将事件流分块(cluster,每个块是若干“事件序列”的集合),再通过比较一个块中的各事件流的属性值生成模式及激励,完成用户模型抽取,并生成结果事实(包括一个 if-then 规则,直接由激励转化而成)^[7]。

这种学习方式具有域独立性及高效增量式学习的特征。另外由于它提供了一种语言来描述系统,分块算法也可以配置,使得用户可以控制学习的内容及方式,学习引擎的健壮性、可扩展性也非常好。

结束语 以上介绍了信息 Agent 的一般结构及算法。解决网上 Agent 系统的开发问题一般都在尽量利用已有的软件系统,从而在保证兼容性的基础上,采用不同的 Agent 的建模方式。上面介绍的实现方法一般仍是从传统的面向应用及信息处理考虑建模,模型本身对 Agent 的自主的特性体现得不十分好,我们已经着手用 BDI 模型构建信息 Agent,从 Agent 的精神状态入手考虑建立 Agent 的模型。对于当前界面 Agent 的一般评价标准为:

自主性:Agent 的行为对用户是否透明,是显式学习还是隐式;

用户控制:可信度有多高,可以被忽略或拒绝吗;

特征识别:Agent 帮助用户时用什么特征;

学习:学习方式是实时还是周期性的。

综合考虑上述因素,学习功能是其自主性的有力

信息检索

信息资源

Internet网

智能中介服务

24

86-90

信息检索的智能中介服务

The Intelligent Mediation Services of the Information Retrieval

谭小川

(湛江师范学院数学系 湛江 524048)

G354.4

TP393

Abstract Based on the intelligent mediation services of the information retrieval on the fundamental theory, model construction, effective mediation services and mediation services maintenance, the paper gives analysis and introductions in detail. And the paper also introduces a method for relaxation query in the intelligent mediation services of the information retrieval.

Keywords Information retrieval, Mediation services, Relaxation query

1 引言

在英特网高速发展的今天,面对网上的海量信息,传统的信息检索方法(如关键词法)显得过于笨拙,即使是目前专家们普遍推崇的基于内容的检索方法,仍缺乏实用性。近年来,为了克服信息检索中的种种困难,中介服务的概念应运而生。所谓中介服务是指一个自动化程度相当高的软件,不仅能根据用户的需求从大量文献中收集所需信息,而且能对收集到的信息进行加工、提炼,把真正有用的信息提供给用户。因此,中介服务软件是一种极为重要的中间件。

斯坦福大学的 Wiederhold 等人指出^[1],中介服务有四项主要任务:①从多样化的信息资源中访问和检索相关信息;②把检索到的数据进行抽象并转换为一种通用的表示;③把已经通用化表示的数据通过匹配关键词加以集成;④通过抽象来约简集成化数据以提高信息传播的密度。

图1显示了中介所需完成任务的三个层次。在一个中介结构系统中,转换数据为信息是一个增值的过程,中介服务的具体功能至少应包括以下一些方面的内容:

- ①应用以前提取的索引,快速抉择有关资源;
- ②使用包装程序以处理遗产性资源;
- ③发现源资料中的相关性;
- ④优化访问策略以减少响应时间或降低费用;
- ⑤使用安全过滤保护私人数据;
- ⑥解决领域术语和用词上的差异;
- ⑦发现不适用的资料;
- ⑧使用内插或外插以消除时序数据的差异;
- ⑨约简历史数据为有限快照;
- ⑩把材料抽象为集成所要求的粒度层次;
- ⑪根据连接关键词从不同的源资料集成信息;
- ⑫省略重复信息;
- ⑬评估不同来源的资料的质量;

谭小川 讲师,主要研究方向:计算机应用,计算机网络技术及应用。

支持,也是其智能性、适应性的集中体现。此外为使 Agent 更加适应网络应用,为用户提供更加高质量的服务,因此有必要结合网上应用特点对开发更高效的学习算法进行深入研究。

参考文献

- 1 Cheung D W, et al. Discovering user access patterns on the World Wide Web. Knowledge Based System, 1998, (10): 463~470
- 2 Nwana H S. Software Agents: an overview. The Knowledge Engineering Review, 1996, 11(3): 205~244
- 3 Goldman C V, et al. Musag: An Agent That Learns What

You Mean. 1997

- 4 Moukas A. Amalthaea: Information Discovery And Filtering Using A Multiagent Evolving Ecosystem. Applied Artificial Intelligence, 1997(11): 437~457
- 5 Payne T R, Edwards P. Interface Agents That Learn. An Investigation of Learning Issues in A Mail Agent Interface. Applied Artificial Intelligence, 1997(11): 1~32
- 6 Schrooten R, et al. Software Agent Foundation for Dynamic Interactive Electronic Catalogs. Applied Artificial Intelligence, 1997(11): 459~481
- 7 Caglayan A, et al. Learn Sesame——A Learning Agent Engine. 1997
- 8 Yager R R. Intelligent Agents for World Wide Web Advertising Decisions. International Journal of Intelligent Systems, 1997, 12: 390~397