基于多 Agent 技术的自动文摘系统的研究和设计

胡舜耕,刘晓宇,钟义信
(北京邮电大学信息工程系,北京 100876)

摘 要: 为了解决自动文摘系统目前所面临的领域通用性和文摘质量的矛盾,提出了建造基于多 Agent 技术的自动文摘系统的方案. 建造这样的系统时,有两个关键问题:首先是确定在一定负载下,面向各个领域的合适的文摘 Agent 数目,其次就是选择什么样的协调算法. 本文给出了在 Internet 环境下基于多 Agent 技术的自动文摘系统模型,提出了三种协调算法,在仿真的基础上分析了系统的性能,得到了在一定负载下面向各个领域的合适的文摘 Agent 数目,并对三种协调算法进行了比较研究.

关键词: 自动文摘; 多 Agent 系统; Agent 数目; 协调算法; 仿真; 分析
中图分类号: TP18 **文献标识码**: A **文章编号**: 0372-2112 (2001) 02-0247-03

Research and Design of Automatic Abstracting Systems Based on Multiagent Technologies

HU Shun-geng, LIU Xiao-yu, ZHONG Yi-xin
(*Department of Information Processing, Beijing University of Posts and Telecommunications, Beijing* 100876, *China*)

Abstract: To resolve the contradiction of domain currency and abstract quality of automatic abstracting systems, we put forward the solution to build automatic abstracting systems based on multiagent technologies. To build such systems, there are two key problems: firstly, how many abstracting agents for each domain are suitable in some load. Secondly, what coordination strategy should be used. In this paper, the model of an automatic abstracting system based on multiagent technologies is given, and three kinds of coordination algorithms are set forth. We analyze the performance of the system based on the simulations, and get suitable number of abstracting agents for each domain in given load. Furthermore, we compare three given coordination algorithms.

Key words: automatic abstracting; multiagent system; agent number; coordination algorithm; simulation; analysis

1 引言

多 Agent 系统是一个松散耦合的 Agent 网络,这些 Agent 通过合作解决超出单个 Agent 的能力或知识的问题. 其中的 Agent 是自主的,它们可以是不同的个人,基于不同的平台,采用不同的设计方法和计算机语言开发而成的,因而可能是完全异质的. 多 Agent 技术被认为是复杂的,开放的分布式问题求解的一种可行的解决方案[1]. 随着 Internet 的迅猛发展,多 Agent 技术在建造 Internet 环境下的各种各样的应用系统中得到了越来越广泛的应用,如日程安排、信息获取、远程教育、电子商务和自然语言理解[2,3]等. 本文主要讨论在 Internet 环境下建立基于多 Agent 技术的自动文摘系统的若干问题.

自动文摘系统所面临的适用领域和文摘质量的矛盾的一种可行的解决方案就是建造基于多 Agent 技术的自动文摘系统[2]. 说它可行,一是多 Agent 理论和技术的发展为建造实用的多 Agent 系统提供了可能性,二是 Internet 的飞速发展为多 Agent 系统的建造提供了天然的平台. 再者,Internet 的快速发展也呼唤网络自动文摘服务.

本文给出了在 Internet 环境下基于多 Agent 技术的自动文摘系统模型,提出了三种协调算法,在仿真的基础上分析了系统的性能,得到了在一定负载下面向各个领域的合适的文摘 Agent 数目,并对三个协调算法进行了比较研究.

2 系统模型

设计的基于多 Agent 技术的自动文摘系统能向 Internet 用户提供自动文摘服务. 如图 1 所示,系统包括一个动态变化的用户集,它们通过 Internet 向系统请求自动文摘服务. 其中标识为 SA(Scheduling Agent)的是协调者 Agent,它负责文本分类和系统的动态任务分派,也是用户界面 Agent. 标识为 IA(Information Agent)的是信息 Agent,它负责文摘 Agent 的注册登记、身份认证及撤消注册等工作. 标识为 abs_{11}, …, abs_{1k_1}, …, abs_{n1}, …, abs_{nk_n} 的是文摘 Agent,它们在后台完成自动文摘服务. 所有的文摘 Agent 称为一个文摘 Agent 网络(AAN,Abstracting Agent Network). abs_{i1}, …, abs_{ik_i} 称为 AAN 中的第 i 类文摘 Agent,或第 i 类文摘 Agent 子网络. abs_{ij} (abs 指 Abstracting

收稿日期:2000-01-25;修回日期:2000-09-10
基金项目:国家自然科学基金(No. 69982001)

System)称为第 i 类文摘 Agent 子网络中第 j 个文摘 Agent,而且规定序号越小,文摘 Agent 做出的文摘质量越高.这里,$i \in \{1,2,\cdots,n\}$ 是文摘 Agent 网络所适用的领域标号,k_i 是面向领域 i 的文摘 Agent 数目.文摘 Agent 可以由不同的组织,基于不同的平台,使用不同的设计方法开发而成,可能是完全异质的.它们所作出的文摘质量有高低,服务时间有长短.

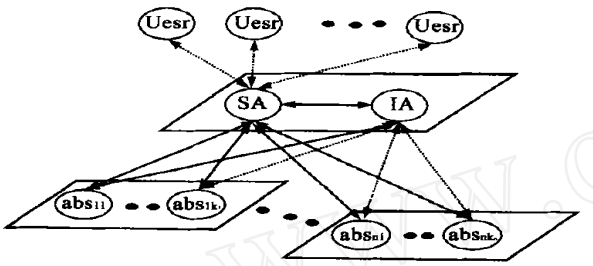

图 1 系统模型

建立用户或文本模型的一个关键问题是用户向系统提交的文本是如何在不同领域之间分布的,即提交的文本属于某一个领域的概率有多大.采用纯 Zipf 函数来模拟文本在不同领域之间的分布.Zipf 形分布是这样定义的:对一个有 L 个元素的集合,选择第 i 个元素的概率 q_i 定义为 $q_i = c/i^{(1-x)}$.这里 $c = 1/[\sum_{i=1}^{L} 1/i^{(1-x)}]$ 是归一化因子[4~6],$0 \le x \le 1$ 为参数.所谓纯 Zipf 函数就是参数 $x = 0$ 的情形.

系统的工作原理是这样的:用户把文本通过适当方式(如 E-mail)提交给协调者 Agent SA ,SA 从信息 Agent IA 处获取在线文摘 Agent 姓名和地址,经过投标,SA 根据预先确定的协调策略决定把文本提交给合适的文摘 Agent.该文摘 Agent 负责完成文摘任务以后,再把文摘传送给 SA.最后 SA 把文摘以适当方式提交给用户.若用户所提交的文本超出系统的适用领域,它会拒绝提供服务,并给出提示.

3 协调策略

当 SA 接收到一个用户提交的文本时,它如何决定把该文本提交给一个合适的文摘 Agent? SA 不仅利用环境信息即用户或文本信息,而且也可以利用 AAN 的信息,完善的协调策略还将考虑文摘 Agent 的历史信息.环境信息即文本所属的领域和文本长度等.AAN 信息就是 AAN 中各个文摘 Agent 的队列长度或服务时间等.文摘 Agent 的历史信息主要指在过去一个时间段内的利用率.下面将给出三种协调算法.

3.1 最小队列算法(Min Q, Min Queue)

这种协调算法是根据文本所属领域和 AAN 中各个文摘 Agent 的队列长度来决定提供服务的文摘 Agent.协调者 Agent 首先确定用户提交的文本所属领域 TD(Type of Domain,领域类型),然后将文本提交给第 TD 类文摘 Agent 子网络中当前队列最短的文摘 Agent 中序号最小的 Agent.如果该类文摘 Agent 子网络中所有文摘 Agent 的队列均满,用户的文摘请求将被拒绝.

3.2 最小等待时间算法(MinWT, Min Wait Time)

一般而言,AAN 中的文摘 Agent 是异质的,对相同文本的服务时间是不相同的.MinQ 算法没有考虑文摘 Agent 的服务时间.事实上,可能出现这样的情况,一个文摘 Agent 尽管其当前队列很长,但因为服务速度快,所以等待中的文本能很快被服务.相反,如果一个文摘 Agent 的当前队列很短,但服务很慢,那么文本等待服务时间就长.所以,在一个高度异质的多 Agent 系统中,还应该考虑服务时间问题.

利用 MinWT 算法,协调者 Agent 首先确定用户提交的文本所属领域 TD,然后将文本提交给第 TD 类文摘 Agent 子网络中等待服务时间最短的文摘 Agent 中序号最小的 Agent.如果该类文摘 Agent 子网络中所有文摘 Agent 的队列均满,用户的文摘请求将被拒绝.

3.3 考虑历史信息的算法(HIMin Q, History Information and Min Queue)

上述两种算法既考虑了来自环境或用户的信息,又从不同方面考虑了 AAN 中文摘 Agent 的当前状态,但都没有考虑它们的历史信息.本节介绍综合考虑历史信息的算法.

利用 HIMinQ 算法,协调者 Agent 首先确定用户提交的文本所属领域 TD,然后将文本提交给第 TD 类文摘 Agent 子网络中在提交文本时刻过去 240 秒内文摘 Agent 的利用率和该时刻的归一化队列长度的加权平均最小的文摘 Agent.如果该类文摘 Agent 子网络中所有文摘 Agent 的队列均满,用户的文摘请求将被拒绝.

4 仿真结果和结论

在 Unix 支持下的 BONeS 仿真环境下建立了离散事件的仿真模型.在实际系统中,文摘 Agent 是异质的,解决问题的能力是各不相同的.但在我们的仿真模型中只考虑了一种特殊情况,即面向同一领域的文摘 Agent 能力是相同的,无差别的.具体表现在面向同一领域的文摘 Agent 所作出的文摘质量相同,对同一长度文本作出文摘的服务时间相同.选择的度量 AAN 性能的指标是:文摘 Agent 中最大队列长度;文摘 Agent 的平均队列长度;文摘 Agent 的利用率;文摘 Agent 子网络的系统利用率.通过仿真,得到了在前述三种协调算法下,面向各个领域的文摘 Agent 数目从 1 到 10 的各种情况下的上述 AAN 的性能指标.仿真结果略.

表 1 面向各个领域的文摘 Agent 数

领域编号	1	2	3	4	5	6	7	8	9	10
文摘 Agent 数	6	4	3	3	2	2	2	2	2	2
平均利用率(MinQ)	.4767	.3554	.3184	.2355	.2835	.2366	.2000	.1785	.1607	.1416
平均利用率(MinWT)	.4694	.3505	.3128	.2319	.2800	.2335	.1982	.1766	.1575	.1397
平均利用率(HIMinQ)	.4694	.3505	.3128	.2319	.2800	.2335	.1982	.1766	.1575	.1396
最大队列长度	2	2	2	2	2	2	2	2	2	2

利用的确定面向各个领域的合适的文摘 Agent 数目的原则是:(1)在三种协调算法下,每个文摘 Agent 子网络的平均利用率都不超过 0.5.所谓文摘 Agent 子网络的平均利用率就是文摘 Agent 子网络的系统利用率的期望值.(2)每个文摘 Agent 中的最大队列长度为 2.依据上述原则,得到面向各个领域的合适的文摘 Agent 数目,见表 1.

关于协调算法,仿真结果表明,如果只考虑文摘质量而不

关心任务分配的均衡性,利用算法 MinQ 较好,它所需系统信息也是最少的.如果既要考虑文摘质量又要任务分配较均衡,算法 MinWT 较好,但它所需系统信息也较多.在三种算法中,利用 HIMinQ 算法时,面向同一领域的各文摘 Agent 所分配的任务量和工作时间都基本一致.如果面向同一领域的各文摘 Agent 所作出的文摘质量相差不大,利用 HIMinQ 算法是一个好的选择.当然,HIMinQ 算法需要获取历史信息,有一定的系统开销.

本文的工作为在 Internet 环境下设计和开发基于多 Agent 技术的自动文摘系统提供了依据.而且,只要对诸如系统适用的领域数量,系统的访问量或负载等参数作适当的修改,可以得到更多具有指导意义的仿真结果.正是在上述工作的基础上,利用北京邮电大学智能研究中心研制的的几个基于单机的自动文摘系统,在 Internet 环境下建造了一个基于多 Agent 技术的自动文摘系统.协调算法 MinQ 注重文摘质量,它所需文摘 Agent 网络信息也是最少的,实现较简单.因此,系统目前采用的是 MinQ 协调算法.

鸣谢:感谢刘伟权,杨晓兰,李蕾,孙春葵和郭祥昊博士,他们研制了基于单机的自动文摘系统.感谢张莉和魏强同学,他们为基于多 Agent 技术的自动文摘实验系统的开发花费了大量的心血.

参考文献:

[1] Sycara K. P. Multiagent systems [J]. AI magazine, summer 1998:79 - 92.

[2] Hu Shungeng, Zhong Yixin, Wei Chaocheng. An automatic abstracting architecture based on multiagent technologies [A]. Proceedings of International Conference on MT & CLIP [C], June 1999. Beijing, China.

[3] Stefanini M. H., Demazeau Y. TALISMAN: a multi-agent system for natural language processing [J]. LNAI (Lecture Notes in AI), 1995, 991:312 - 322.

[4] Zipf G. K. Human Behaviour and the Principles of Least Effort [M]. Cambridge, Mass.: Addison-Wesley, 1949.

[5] Colajanni M. et al. Analysis of task assignment policies in scalable distributed web-server systems [J]. IEEE Trans. on Parallel and Distributed Systems, June 1998, 9(6):585 - 600.

[6] Cunha C. et al. Characteristics of WWW client-based traces [R]. Technical Report BU-CS-95-010, Computer Science Dept., Boston Univ., Apr. 1995.

作者简介:

胡舜耕　1964 年生,1989 年毕业于四川大学数学系,获硕士学位,1997 年在北京邮电大学信息工程系攻读博士学位至今.主要研究方向为多 Agent 技术、中文信息处理、Internet 信息服务.

刘晓宇　1972 年生,1995 年毕业于北京邮电大学电信系,1997 年在北京邮电大学信息工程系攻读博士学位至今.主要研究方向为移动通信系统建模、仿真和性能分析.

钟义信　1940 年生,教授,博士生导师.主要研究方向为信息科学、人工智能、神经网络.