Internet 网上智能检索 Agent 的实现方法

刘树勋 李卫华 区益善 （广东工业大学计算机系 广州 510090）

摘要：论述了一种 Internet 网上智能检索 Agent 的取词方法。Agent 提取用户提交的 WEB 文档的关键短语，通过机器学习，产生决策树，然后生成布尔查询串，从而学习到用户的兴趣，自主地在 Internet 网上进行信息检索。

关键词：Internet 网 智能检索 Agent 机器学习

随着 Internet 网络的迅速发展，各种类型的信息服务层出不穷，许多技术问题的解决亟需智能软件技术的帮助。传统的检索引擎，如：Altavista，Webcrawer，Infoseek，Lycos，Yahoo 等都是服务器端软件，用户需严格按照各个引擎所要求的格式输入查询串。这类引擎不具备智能性，不能学习用户兴趣，查询准确性也不高。如果把检索工具与查询数据库分离，把检索工具安置在客户端，并且利用机器学习技术，使客户端检索软件具备智能性，能够学习用户兴趣，就能弥补传统引擎的不足。这种检索工具被称之为 Internet 网上智能检索 Agent。对于一给定的主题，如何利用 Agent 在网上检索相应的文档，类似这方面的研究，已经引起了很多很多科研人员的重视。目前关于 Agent 的研究无论是在 AI 界，还是在计算机科学的其他领域，都十分活跃。机器学习技术对提高 Agent 的智能有着密切的作用。把机器学习技术应用到 Agent 中，是 AI 研究领域的一个新的发展方向。

在我们即将实现的智能检索 Agent 中，采用了一种较好的 WEB 文档取词方法，并且利用了机器学习技术，我们称之为 InfoSearcher Agent。用户只要提交自己感兴趣的 WEB 文档给 InfoSearcher Agent，它就能利用机器学习技术学习到用户的兴趣，并且自主地在 Internet 网上漫游，收集用户感兴趣的信息。每个用户都能按照自己的习惯配置 Agent，使之具有个人的独特风格和识别特定的语义模式的能力，提高了 Agent 的灵活性、准确性、自主性、智能性。InfoSearcher Agent 可以十分灵活地提供多种智能化的信息处理手段，将有力地开拓 Internet 网络上的信息服务。

1. 学习用户兴趣

InfoSearcher Agent 从用户提交的 WEB 文档中学习用户兴趣。当用户提交 WEB 文档给 InfoSearcher Agent 时，它通过启发式短语提取方法，提取 WEB 中的关键短语。提取文档关键短语后，通过改进的 ID3[1] 算法学习用户的兴趣。为了使 Agent 能真正学习到用户的兴趣，要求用户在提交 WEB 文档给 Agent 时，按照用户不同的兴趣，对 WEB 文档进行分类。代表用户不同兴趣的文档放在不同的文件夹下，Agent 通过对不同文件夹下的文档的学习，从而学习到用户不同的兴趣。为了提高 Agent 学习用户兴趣的准确程度，对不同的兴趣，一般要求用户提供 10 篇以上的样本文档给 Agent，Agent 按照一定的推理规则对文档逐篇学习。Agent 通过三个步骤实现这个学习过程：

(1)用启发式短语提取方法，逐篇提取 WEB 文档的关键词；

(2)在提取关键词的基础上，通过改进的 ID3 算法产生一棵决策树；

(3)把决策树转换为一个布尔查询串。

InfoSearcher Agent 依靠观察文档作者阐述文档中关键短语的方式来实现启发式短语提取方法。如：作者把关键短语置为斜体、缩写、加着重号等。为了简便，InfoSearcher Agent 只从文档标题中提取关键短语。在以后的推理过程中，那些不能反映文档主题的短语将被 Agent 抛弃掉。

```
JAVA
    Applet,c++,client/server,guide,IBM,Internet,Java,javascript.....(正例)
    IBM，ATM,directory,Internet,Java,LEC,EDI.....(反例)
    IBM，Java,Microsoft windows,NetScpae.....(正例)
    Client/server,server inteaddress,3-D.....(反例)
    3GL,c++,client/server,CORBA,Internet,Java,javascript.....(正例)
```

图 1 JAVA 文件下的样本文档

当用户提交 WEB 文档后，InfoSearcher Agent 就利用用户提交的文档学习用户的兴趣。提交的文档越多，学习效果越好。在图 1 中示出了“JAVA”文件下的有关文档，每一篇文档都由用启发式短语提取方法所取出来地短语代表。

提取关键短语以后，Agent 就可以利用改进地 ID3 算法来学习用户兴趣。虽然有的研究人员宣称 ID3 算法不如其他学习算法效率高，但高效的启发式短语提取方法弥补了 ID3 算法效率底的不足。同时我们还改进了 ID3 算法，提高了其学习效率。图 2 显示了 JAVA 文件下的决策树。这棵树显示了 Agent 已经抓住了 JAVA 文件下文档的重点，而不是文档中无关紧要的内容。而且 InfoSearcher Agent 需使用的文档数目并不多，不象有些取词算法，需要许多篇文档才行。

当 InfoSearcher Agent 已经归纳出一棵决策树，它就产生一个布尔查询串，如图 2 中的布尔查询串，这种转换比较简单。系统然后就利用这个布尔查询串在 Internet 网上查询与用户兴趣相符的文档，并且把查询到的文档发送给用户。用户对查询到的文档作出评价，告诉 Agent 哪些文档与用户兴趣相符，哪些相异。Agent 根据用户的反馈，更新布尔查询串，使之更符合用户的意图。显然，这个学习过程是交互式的，用户需适时地提供反馈信息给 Agent。这种选择和精练样本文档的交互过程，其实就是对 Agent 的归纳算法提出了某种暗示，告知其学习到的知识哪些是正确的，哪些是错误的。尽管大部分学习系统希望用户提供的文档是正面例子，但 InfoSearcher Agent 希望用户在提供正例的同时，也提供反例，使其能更好地学习到用户的兴趣。

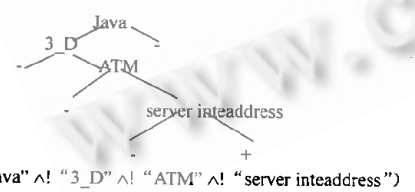


图 2　由 ID3 算法产生的 JAVA 文件夹下的决策树

2. 从文档中提取关键词

学习过程中最重要的一步是关键词的提取。对于学习算法本身，不管采用多么好的模式匹配方法，系统只有接受高质量的关键词，模式匹配算法才能发挥它的应有效率。以前的研究人员总是尽量处理整篇文档，但基于以下三个原因，我们避免处理整篇文档，而只是抽取文档中的一部分来提取关键词：

(1)WEB 文档有一定的结构，在查询时利用这种结构信息，将显著提高查询的准确性。可要求关键词出现 title、head、text、site、URL 等特定的地方；

(2)文档中只有少数短语反映了文档的重点，而且分散的短语不能反映文档的本质内容；

(3)处理整篇文档代价太高，对于大型文档更不可采用这种整篇处理的方法，只从语义上提取重要短语处理起来就比较简单。

当 Agent 打开一个 WEB 文件时，它读取文件的 title、head、text 等元标识对之间的内容。同时，还允许用户使用元标识“keywords”为文档提供附加的索引信息，用元标识“description”为文档建立摘要。Agent 应该读取这些元标识对之间的信息。提取这些元标识对之间的信息以后，再在这些信息中提取关键词。

在相关研究领域中[2,3,4]，有一种从文档中提取可视化特征的方法，InfoSearcher Agent 的启发式短语提取方法就是基于这种文档可视化特征提取方法的基础之上的。这种方法依靠观察文档作者采用哪几种可视化方法向读者表达文档重点。如斜体、加着重号、把重点条款列于表格之中、文档结构、摘要、逻辑表达式等。识别这些可视化特征就可以让 Agent 从文档中提取关键短语。

例如：提取文档中以大写字母书写的词这样一个简单的启发式取词算法。这样的词很可能是缩写词专业技术名词等。而且有许多方法可以找到这个缩写词的定义，如搜寻紧跟在缩写词后面括号中的内容；如果这个缩写词本身是用括号括起来的，则可以在缩写词前查找这个缩写词的定义。

另外一种简单的启发式取词算法是提取与周围文档格式不同、而且不是一个完整句子的短语，这种短语一般由一至五个单词组成。利用人们在首次使用关键短语时一般会大写或加下划线或大写第一字母的习惯，就可以实现这种算法。

这一类启发式取词算法还包括列表的识别、段落标题的识别、图表的识别、高重复率短语的识别等。对于这些算法，我们还应该建立两个数据库：一个用来存储那些并不重要的短语，如：not、TM、certainly 等；另一个用来存储同义词。当 Agent 提取到这些词时，就应该主动抛弃

掉。

对提取出的所有词按其在文章中的每个位置打分，将各个位置的分数累计，按总得分多少排序，总得分多的就认为是关键词了。

3. 改进的 ID3 算法

InfoSearcher Agent 的学习算法要求用户对样本文档进行分类，这与传统应用领域的机器学习方法不同。上面我们已经讨论过，InfoSearcher Agent 并没有提供给用户一系列文档，以调查用户对哪方面的信息感兴趣，而是当用户浏览 WEB 文档时，用户选择文档提交给 Agent，作为样本文档。用户一般很少会提交与他们兴趣相异的文档给 Agent，而是提交正面的文档来训练 InfoSearcher Agent。另外用户也可以提交反例(即与用户兴趣相异的文档)，以消除 InfoSearcher Agent 所学习到的知识(即它认为用户感兴趣的信息)与用户真正所期望的信息之间的差异。同样，当用户一边浏览一边把 WEB 文档提交给 InfoSearcher Agent 时，文档将会被处理。InfoSearcher Agent 将从用户时期提交的若干篇文档中初步归纳出用户的兴趣范围，并且把学习结果提交给用户，让用户了解 InfoSearcher Agent 学习到的知识与用户真正兴趣之间的差异，然后，用户再提交几篇文档给 InfoSearcher Agent，进一步消除用户与 Agent 之间认识上的差异。这样，早期的文档将使 Agent 对用户的兴趣有一个大致的了解，后面提交的文档将使 Agent 进一步消除对用户兴趣的误解。

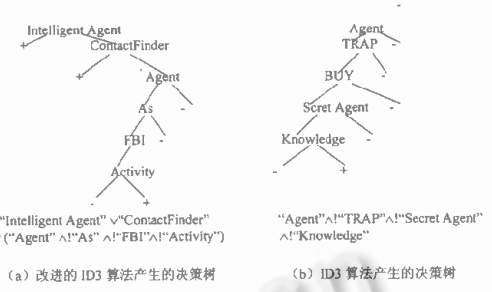
InfoSearcher Agent 的归纳算法使用用户提交的头 10 篇文档来归纳用户兴趣的大致范围，再利用后续文档来校正这种归纳。当用户提交的文档还不到 10 篇时，InfoSearcher Agent 每收到一篇文档，就采用 ID3 算法，从头逐篇学习。当用户提交的文档超过 10 篇时，InfoSearcher Agent 就采取增量归纳法，即简单的抛弃掉原来的文档，不再从头学习，只把新文档融合到以前产生的决策树中。在理论上，如果文件夹下包含的文档数量较多，尤其是当文档数量超过 10 篇以后，用户后面提交的文档所涉及的内容与前面提交的文档内容差别较大时，就应该重构决策树，否则算法的学习效果就会很差。我们之所以采用这种改进的 ID3 算法，是基于以下两个原因：

(1)提高 ID3 算法的学习效率；

(2)在同一文件夹下，我们假设用户提交的后续文档与前面提交的文档所涉及的内容差别不大，即假设在同一文件夹下用户的兴趣没有改变。在现实生活中，这种假设是可以的，因为用户同一文件夹下一般不会改变自己的兴趣。

在实验中，我们提交了 40 篇有关智能 Agent 的文档给两个不同版本的 InfoSearcher Agent，一个采用改进的 ID3 算法，当用户提交的文档超过 10 篇以后，对于后续文档，采用增量归纳法，不重构决策树；另一个只采用 ID3 算法，用户每提交一篇文档给 Agent，它就从头开始学习，重构决策树。采用改进的 ID3 算法的 Agent 的学习效率比采用 ID3 算法的 Agent 的学习效率高出一倍以上。图 3 就是这两个不同版本的 Agent 所产生的决策树，两个决策树具有相似的结构，这说明采用改进的 ID3 算法的 Agent 的学习效果是可以的。

当 Agent 已经归纳出一棵决策树，它就产生一个布尔查询串，然后提交给普通的搜索引擎。这种转换比较简单，只需从树的根结点到叶结点顺序转换就可以了。



(a) 改进的 ID3 算法产生的决策树　(b) ID3 算法产生的决策树

(a) 改进的 ID3 算法产生的决策树

(b)AgentAgentAgent ID3 算法产生的决策树

图 3

4. 网络漫游

为了限制 InfoSearcher Agent 在 Internet 网上的漫游范围，我们限制 InfoSearcher Agent 在网络上的搜索深度为 3。网络搜索的算法如下：

(1)初始化两个链表：currentHTMLURL，netHTMLURL。链表 currentHTMLURL 存储当前 WEB 文档中的超链，链表 nextHTMLURL 存储由 currentHTMLURL 链表中的超链所对应的 WEB 文档中的超链；

(2)分析当前 WEB 文档，提取其中的超链，经过 URL 有效性验证，且 currentHTMLURL 中没有此 URL，则将 URL 插入链表 currentHTMLURL 中；

(3)While(currentHTMLURL<>NULL)，则

{

连接链表中 URL 所对应的 WEB 主页，并取回本地，对文档进行分析，

if(符合用户兴趣)then{

把 HTML 保存在本地，且提取 HTML 中的 URL，经有效性验证，且 nextHTMLURL 中没有此 URL，则将此 URL 插入 nextHTMLURL 中

}

else

{

否则抛弃此文档

}

};

(4)While(nextHTMLURL<>NULL)

{

连接链表中 URL 所对应的 WEB 主页，并取回本地，对文档进行分析

if(符合用户兴趣)then

{

把 HTML 保存在本地

else

{

抛弃此文档

}

}

需要指出的是，InfoSearcher Agent 主动放弃对以下网址的访问：

(1)需要用户输入用户名和口令才能继续浏览的 WEB 主页；

(2)URL 所对应的不是 WEB 主页，如是 FTP、Telnet 和电子邮件等；

(3)由于网络过于拥挤、目标 URL 地址错误、目标网络无法到达等原因，引起下载超时的 URL 地址。

5. 结束语

正如我们上面所描述的，Agent 通过一个动态的交互过程来学习用户兴趣。尽管在这个领域大部分的研究集中在归纳算法上，但为了兼顾 Agent 的学习效率与学习准确性，我们在设计 Agent 时，采用了较好的取词算法，同时为了提高学习的效率，采用了改进的 ID3 学习方法。这样，有助于 Agent 在花费较小代价的情况下，获得较好的学习效果。

进一步的研究我们将集中在以下几个方面：

首先，我们希望找到效率更高，准确性更高的取词算法。虽然在具有一定格式的 WEB 文档上，前面所说的取词算法工作得较好，但在没有一定格式的文档中却效率不高，尤其是文档篇幅较大时，效率更低。我们希望找到在其他无一定格式的文档中也工作得较好的取词算法，以拓展 Agent 的应用范围。

其次，应该使 Agent 运行在工作群中，使一个工作群中有相同兴趣的成员能够共享他们共同感兴趣的信息，允许 Agent 从其他有共同兴趣的成员的文件夹下读取文档，做到信息共享。

参考文献

[1] R. Quinlan, learing Efficient Classfication Procedure and Their Application Chess And Game, Tioga Publishing, Palo Alto, Calilf, 1983, 67～85

[2] B. Krulwich and C. Burkey, The ContactFinder Agent: Answering Bulletion Board Question with Referrals, Proc Nat'l Conf. AI, AAAI Press, 1996, 10～15

[3] K. Swaminathan, Tau: A Domain Independent Approach to Information Extration from Natural Language Document Management, Palo Alto, 1993, 102～109

[4] M. Pazzani, J. Muramatsu, and D. Billsus, Syskill & Webert: Identifying Interesting Web Sites, Proc. Nat'l Conf. AI, AAAI Press, 1996, 51～61

[5] Chud Burkoy, Anderson Consulting LLP. IEEE EXPERT, 1997(9), 22～27

(来稿时间：1999 年 4 月)