基于 Agent 的智能元搜索引擎技术研究*)

李红梅[1,2]　丁振国[1]　周利华[1]
(西安电子科技大学计算机学院　西安 710071)[1]
(河北农业大学信息科学与技术学院　保定 071001)[2]

摘　要　针对现有搜索引擎存在的问题,提出基于 Multi-agent 的分布式搜索引擎系统。系统采用元搜索引擎结构,利用 Agent 技术和基于个性化模式的信息过滤技术,使系统具有一定的智能性。通过个性化检索和分类浏览相结合的检索方式可提高搜索结果的可浏览性。结合数据库的分类和虚拟语言模型方法实现了资源选择的优化。提出基于文本/位置分析和群决策的合并算法,对搜索结果的标题和文档片断信息进行相关度分析,将文本分析与规范化的搜索结果位置信息相结合,计算文档的相关分值,最后采用基于群决策的合成方法对搜索结果进行一致性排序。试验结果表明,提出的元搜索系统具有较好的搜索效果。

关键词　信息检索,Agent,元搜索引擎,个性化检索

Research on Intelligent Metasearch Engine Based on Agent

LI Hong-mei[1,2]　DING Zhen-guo[1]　ZHOU Li-hua[1]
(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)[1]
(College of Information Science and Technology, Agricultural University of Hebei, Baoding 071001, China)[2]

Abstract　A intelligent metasearch engine system based on multi-agent is proposed. The agent technique and information filtering technique based on personalized models are utilized, which makes the system more inteligent. The retrieval methods combining customized search with classified browse help users find relevant results more quickly. The scheduling of search sources is optimized by integrating the database categorization with virtual language model approach. A result merging method based on text / rank analysis and group decision making activity is presented. By utilizing text-based information such as title and snippets obtained from search results, the method to analyze the relevancy of title and snippets is described. Then, the relevant scores of the relevant documents are normalized by incorporating text analysis together with rank. Finally, a merging method based on group decision making activity is adopted to sort the search results. The experimental results show that this system has a better performance.

Keywords　Information retrieval, Agent, Metasearch engine, Personalized retrieval

1　引言

由于互联网信息呈指数增长,而单一搜索引擎网络资源的覆盖率不超过整个 Internet 资源的三分之一,同时因为各个搜索引擎的索引技术不同,导致对同一查询请求,不同的搜索引擎查询的结果重复率很低,因此,要想获得一个比较全面、准确的结果,就必须反复调用多个搜索引擎。此外,搜索引擎不能按照用户的需求对搜索结果进行排序,不同用户提交相同的关键字查询请求时,由于其偏好不同,对所需要的信息要求也不同,但搜索引擎返回的搜索结果却是相同的,无法满足用户的个性化需求。

针对以上问题,本文将智能 Agent 技术和元搜索引擎技术相结合,提出了智能元搜索引擎模型。

2　个性化智能元搜索引擎系统模型

2.1　个性化智能元搜索引擎系统的体系结构

设计了一个智能元搜索引擎系统,采用分布式智能体 Agent 技术与元搜索技术相结合,进行并行的查询和检索,并从用户角度出发,基于用户的访问日志和网页点击行为建立个性化模式,对检索信息进行智能过滤,同时采用了个性化检索和聚类浏览相结合的检索方式,既能满足用户的个性化

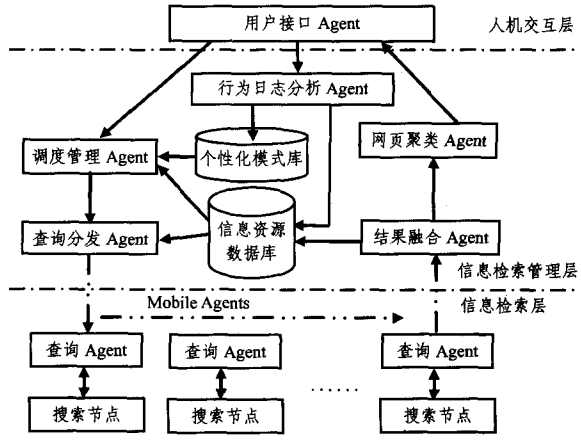

图 1　系统体系结构框架

*)基金项目:国家 863 项目(2004AA1Z2520)。**李红梅**　博士研究生,主要研究方向为信息处理和人工智能;**丁振国**　博士,教授,主要研究方向为计算机网络与信息处理;**周利华**　博士生导师,教授,主要研究方向为多媒体和计算机网络安全。

需求又能适应用户需求的变化。

系统是一个多 Agent 系统，在逻辑上分为三层体系结构：人机交互层、信息检索管理层、信息检索层。其系统体系结构框架如图 1 所示。

2.2 个性化智能元搜索引擎系统的系统功能

2.2.1 人机交互层

它向系统发出请求和接受系统的服务，主要为用户提供一个友好的交互界面，负责接收用户的输入并将检索的结果以分类目录形式显示给用户，方便用户对信息的浏览。

2.2.2 信息检索管理层

信息检索管理层的各个 Agent 的具体功能如下：

(1)调度管理 Agent

调度管理 Agent 根据信息资源数据库中对各个成员搜索引擎的性能评价信息以及用户个性化模式信息，通过调度算法产生一个合适的成员搜索引擎列表。

(2)查询分发 Agent

由于各个成员搜索引擎所支持的查询格式各不相同，查询分发 Agent 在将查询请求转化成对应目标搜索引擎的查询参数格式后发送到查询 Agent 进行信息的检索。

(3)结果融合 Agent

结果融合 Agent 负责接收成员搜索引擎的返回结果，根据搜索结果的合成算法对成员搜索引擎返回的结果进行合并，并进行必要的去重处理。此外，还将成员搜索引擎的负载等状态信息存储到信息资源数据库。

(4)网页聚类 Agent

网页聚类 Agent 对合并后的搜索结果进行聚类分析，创建类目体系，以分类目录形式显示给用户，使用户能在更高的主题层次上来查看搜索引擎返回的结果，便于快速定位到查找信息，方便用户浏览。

(5)行为日志分析 Agent

行为日志分析 Agent 负责对用户行为日志进行分析和挖掘，产生个性化模式，存储在个性化模式库中。用户点击行为也是对成员搜索引擎与查询请求之间相关度的正面评价，对每次查询的用户点击行为进行分析，存入信息资源数据库，对成员搜索引擎的性能评价进行相应的调整。

2.2.3 信息检索层

本文采用移动 Agent 和静态 Agent 协同工作来完成对成员搜索引擎的信息检索任务。

查询分发 Agent 依据成员搜索引擎列表创建一个或多个移动 Agent，将查询请求提交给查询 Agent，移动 Agent 的数目可根据网络实际状况进行调整。查询 Agent 完成检索任务，并将结果表示为统一的格式提交给移动 Agent，把结果返回到信息检索管理层。为了避免结果融合 Agent 处产生瓶颈而降低元搜索引擎的效率，移动 Agent 可以将查询节点的搜索结果并行合并后再提交给结果融合 Agent。

资源选择和搜索结果的合成是元搜索引擎中的两个重要步骤，也是本文主要介绍的内容。

3 调度策略

元搜索引擎的调度策略是为了研究元搜索引擎如何为用户选择数量合适并贴近用户查询需求的成员引擎。调度策略通常包括资源描述和资源选择两部分。资源选择的目的是只将查询请求转发给那些与查询请求相关度较高的成员搜索引擎，避免造成不必要的网络负载及搜索无用成员搜索引擎的代价。成员搜索引擎与查询请求的相关性则通过对该搜索引擎的资源描述信息进行判断。

本文采用了将成员搜索引擎数据库与概念类相关联的技术。当元搜索引擎接收到一个用户查询时，先把查询映射到相关的概念类，然后将与这些映射概念类相关联的数据库作为搜索对象。

3.1 基于概念类的资源描述

3.1.1 概念类的描述

概念类的构建模型采用网易的中文目录分类，各个概念类的描述由出现在该概念类的所有子类中的词项组成。每个概念类的描述实质上是一篇文档，因此概念类的描述集合可以看作是文档集合，各个概念类的描述就可以表示为含有词项和权重的向量。

概念类 C 的描述可以表示为一个 n 维向量

$$C=\{d_1,d_2,\cdots,d_n\}$$

其中，d_i 代表第 i 个词项 t_i 在概念类 C 中的权重，$1\leqslant i\leqslant n$，n 为概念类集合所有的词项数。权重的计算采用基于概念类描述集合的 $tf\times idf$ 权重。

3.1.2 成员搜索引擎的特征描述

建立概念类集合的特征描述后，需要将成员搜索引擎与概念类建立关联。此处借鉴了文献[1]提出了 HASRD(High Average Similarity over Retrieved Documents)方法的基本思想，将概念类的特征描述看作查询，通过计算成员搜索引擎对各个查询的检索文档和查询之间的相似度来实现搜索引擎与概念类的关联。

查询 q 为概念类 C 的特征描述向量，D_E 为成员搜索引擎数据库：

(1)计算概念类查询 q 与数据库 D_E 之间的相似度。

①提交查询 q 到各个搜索引擎数据库 D_E；

②对每个成员搜索引擎返回的前 M 个文档 d，分别计算其与查询的相似度 $\mathrm{sim}(d,q)$；

③计算这 M 个文档的平均相似度作为概念查询 q 与数据库 D_E 的相似度 $\mathrm{sim}(D_E,C)$(即数据库 D_E 与概念类 C 的相似度)。

(2)对每个成员搜索引擎数据库 D_E，根据其与各个概念类之间的相似度对所有概念类进行降序排列，将 D_E 分配给排在前面的 k 个概念类，则成员搜索引擎的特征描述由与其关联的 k 个概念类及其与搜索引擎数据库的相似度组成。

(3)对于每个概念类 C，将针对其查询的所有文档中出现的词项及其权重作为该概念类的特征表示。

3.2 基于虚拟语言模型的资源选择算法

用户提交查询时，先将用户查询映射到概念类，然后利用虚拟语言模型计算查询与成员搜索引擎数据库之间的相关性。

语言模型的检索算法是一种基于概率的检索模型[2]。语言模型方法对于查询和每个文档都建立了一个语言模型，文档和查询相关性的计算可以看作从该文档的语言模型中产生查询的概率。

$$P(Q/D)=\prod_{q\in Q}(\lambda P(q/D)+(1-\lambda)P(q/C)) \tag{1}$$

其中，q 是查询 Q 中的词项，$P(q/D)$ 是查询词项出现在文档 D 中的概率，$P(q/C)$ 是词项 q 出现在文档 D 所属的集合 C 中的概率。λ 是权重参数。$P(q/C)$ 起平滑作用。

借鉴语言模型的思想，建立了虚拟语言模型。对于用户查询 Q，q_c 为其映射的概念类，成员搜索引擎数据库 D_E 可以

看作虚拟文档，则查询和数据库 D_E 之间的相关性可通过以下公式计算

$$P(Q/D_E)=\prod_{q_c\in Q}(\lambda P(q_c/D_E)+(1-\lambda)P(q_c/G)) \quad (2)$$

其中，$P(q_c/D_E)$是查询映射概念类 q_c 与数据库 D_E 之间的相似度，$P(q_c/G)$为概念类 q_c 与所有搜索引擎数据库之间的全局相似度。

根据式(2)计算用户查询与各成员搜索引擎之间的相关度，并据此计算搜索引擎的性能评价得分。同时综合考虑搜索引擎的响应时间、用户偏好等因素，产生一个合适的成员搜索引擎列表。

4 搜索结果合成算法

每个搜索引擎采用不同的相似度计算方法，导致搜索引擎性能的不均衡，从而使得不同搜索引擎返回的文档列表具有不可比性，需要用合理的方式来调整局部相似度。文档列表中包含每个文档的标题和文档片断(snippet)，可以充分利用这些文本信息，计算与用户查询之间的相似度，结合文档的排列位置以平衡搜索引擎之间的差异。张卫丰[3]提出了摘要/位置排序法，但该方法考虑因素太少，算法过于粗略；文献[4]对排列位置和文本信息都进行了规范化处理，但对检索结果中的重叠信息则未做处理，针对上述算法的不足提出了基于文本/位置分析和群决策的合成算法。

将元搜索引擎的检索结果合并到一起的过程主要包含相关分值规范化处理，非相关文档的相关分值的估计，相关分值合并。

4.1 相关分值规范化

4.1.1 排列位置的规范化处理

K 个搜索引擎 $E_1,E_2,\cdots,E_k$ 返回基于查询 *query* 的文档列表 $L_{1q},L_{2q},\cdots,L_{Kq}$，$|L_{iq}|$表示文档列表 L_{iq}所包含的文档数，第 i 个文档列表 L_{iq} 中的第 j 个文档 d_j 的位置用 p_{ji} 表示。由于搜索引擎返回的排序位置不能直接进行比较，我们根据式(1)对其进行了规范化处理，将第 i 个文档列表 L_{iq} 中的文档位置 p_{ji} 用范围在[0,1]内的分值 C_{ji} 来表示[5]：

$$C_{ji}=((|L_{iq}|-p_{ji}+1)/|L_{iq}|) \quad (3)$$

4.1.2 文档标题的规范化

由于标题内容比较短，同一词项多次出现的情况较为少见，因此只考虑文档标题与查询之间的匹配度。

定义 设查询 *query* 有 M 个词项，文本 *Text* 中包含这 M 个词项中的 N 个($N\leqslant M$)，则文本 *Text* 与查询 *query* 的查询匹配度为 N/M。

标题的查询匹配度：

$$P_{title}=\frac{n_{title}}{n_{query}} \quad (4)$$

其中，P_{title}：标题的查询匹配度；n_{title}：标题中出现的查询词项数；n_{query}：查询的词项总数

4.1.3 文档片断的规范化

对于文档片断，除了计算其查询匹配度外，还根据查询词项在文档片断中出现的频率和出现的位置计算其查询相似度。

(1)文档片断的查询匹配度

$$P_{snip}=\frac{n_{snip}}{n_{query}} \quad (5)$$

其中，P_{snip}：文档片断的查询匹配度；n_{snip}：文档片断中出现的查询词项数。

(2)文档片断的查询相似度：

$$S(snip,query)=\frac{1}{n_{df}}\times\sum_{j=1}^{n_{df}}(1-\frac{loc(j,snip)}{len(snip)}) \quad (6)$$

其中，$loc(j,snip)$：查询词项在文档片断 $snip$ 中第 j 次出现的位置；$len(snip)$：文档片断 $snip$ 的长度；n_{df}：查询词项在文档片断 $snip$ 中出现的频率。

4.1.4 综合排列位置和文本分析的相关分值

综合考虑规范化的文档排列位置和文本分析的内容，对其赋予不同的权重，计算加权和，得到相关文档的最终相关分值。

4.2 非相关文档的相关分值的估算

设有搜索引擎 A 和 B，对于某一查询，文档 d 只在 A 的结果列表中出现，则该文档在搜索引擎 A 内为相关文档，它的相关分值为 s_1，而其在另一个搜索引擎 B 内则为非相关文档，需要对其相关分值进行估计。多数方法将非相关文档的相关分值取值为 0，是基于文档 d 存在于搜索引擎 B 中但未被检索到这种假设条件，而这种假设适用于完全相同或近似相同的数据源，对于存在部分重叠文档的数据源，特别是重叠率较小时其适用性变差。

另一种假设条件是文档 d 在搜索引擎 B 的数据库中不存在。基于这一假设，文献[6]提出了 SDM(Shadow Document Method，SDM)方法，认为如果文档 d 在 B 中存在，则很可能具有与 s_1 相近的分值而被检索出来，据此可对非相关文档的相关分值进行估算。

基于 SDM 方法的基本思想，并充分考虑搜索引擎本身的性能，提出了改进的 SDM 方法。

假设有查询 q 及 n 个搜索引擎 $E_i(1\leqslant i\leqslant n)$，文档 d 出现在 m 个搜索引擎的结果列表中，其相关分值分别为 $s_i(1\leqslant i\leqslant m\leqslant n)$，$e_i$ 为包含文档 d 的 m 个搜索引擎 $E_i(1\leqslant i\leqslant m\leqslant n)$的性能评价得分，文档 d 在其余 $n-m$ 个搜索引擎中的相关分值可按照下式计算：

$$S_d=\frac{k}{m}\sum_{i=1}^{m}(e_i\times s_i) \quad (7)$$

其中 k 是权重因子，根据实验和经验设定。

4.3 基于群决策 GDM(Group Decision Making)的合成方法

通过上面的计算可以获得各个搜索引擎返回的文档列表中文档的相关分值，并对文档作为非相关文档时的相关分值进行了估计。综合考虑上述因素，对相关分值进行合并时借用了群决策的思想，即文档作为选择对象，搜索引擎作为专家根据用户的查询标准对文档进行判断及根据它们对标准的偏好进行排序。

搜索引擎对文档的判断矩阵：

$$S=[S_{ji}]_{L_D\times K}$$

其中，行为搜索引擎检索的文档集合 D，$L_D=|D|$，$j=1,2,\cdots,L_D$；列为 K 个搜索引擎对文档的评价得分，$i=1,2,\cdots,K$；S_{ji}为文档集合 D 中第 j 个文档 d_j 在第 i 个搜索引擎中的相关分值。

在进行数据库选择时，对不同的搜索引擎进行了性能评价，评价得分高的引擎作出的评判其置信度较高，由此可构造专家影响力矩阵 $E=[e_i]_{1\times K}$，$i=1,2,\cdots,K$。在每次检索任务完成后，根据检索结果应对搜索引擎的性能评价作出适应性调整。

文档集合 D 的最终评价得分矩阵通过下式计算：

$$Score_D=S\times E^T \quad (8)$$

最后根据评价得分降序对输出文档排序。

5 试验结果

以通用的搜索引擎 Google、百度(baidu)、雅虎(Yahoo)、搜狐(Sohu) 为成员搜索引擎建立了一个中文元搜索原型 MWS,进行了 20 次各类查询主题的实验。由于研究表明用户浏览的平均页面为 2.35,因此本实验中每次查询从各个搜索引擎中选取返回的前 30 条记录作为合成的输入文档。

表 1 搜索引擎的相关性比较(MWS 为基准)

rank	MWS	baidu	google	sohu	yahoo
5	0.7667	0.6667 (−13.0%)	0.8000 (+4.3%)	0.7333 (−4.3%)	0.6667 (−13.0%)
10	0.7633	0.6167 (−19.2%)	0.6833 (−10.5%)	0.6000 (−21.4%)	0.5667 (−25.8%)
15	0.7557	0.6133 (−18.8%)	0.6777 (−10.3%)	0.5083 (−32.7%)	0.5433 (−28.1%)
20	0.7333	0.5757 (−21.5%)	0.6083 (−17.0%)	0.4787 (−34.7%)	0.4833 (−34.1%)
25	0.7267	0.5667 (−22.0%)	0.5500 (−24.3%)	0.4733 (−34.9%)	0.4533 (−37.6%)
30	0.6723	0.5800 (−13.7%)	0.5643 (−16.1%)	0.4847 (−27.9%)	0.4447 (−33.9%)

对搜索结果的评价采用了文献[7]中提出的相关性评价方法:

$$relevancy=\frac{2*r+u}{2*n} \tag{9}$$

其中,r 是相关文档,u 是不确定文档,n 为检索到的文档总数。

实验测试了元搜索原型 MWS 的平均相关性,并与各成员搜索引擎进行比较,其实验结果见表 1 所示。

表 1 为元搜索原型 MWS 与成员搜索引擎的搜索结果平均相关性比较。从表中可以看出 MWS 的检索结果相关性比成员搜索引擎均有较大的提高。对于中文搜索而言,百度和 Google 是两个比较好的搜索引擎,针对不同的搜索主题,其性能各有优劣。在本试验的结果中,Google 的相关性略高于百度。而本文的搜索原型与最好的 Google 相比,除前 5 个文档的相关性略有降低外,平均相关性提高了 10%～20%,比最差的成员搜索引擎提高了 10%～35%。

结束语 本文提出了一个基于 Muti-agent 的元搜索引擎系统。采用了多 Agent 技术和移动 Agent 技术,可以减轻网络负载,且对网络连接的要求不高,提高了元搜索引擎的效率,并具有良好的可扩展性。数据库的分类有利于元搜索引擎中的资源选择算法选择与用户查询更相关的数据库,从而提高检索效率。在搜索引擎的调度策略中,在对搜索引擎数据库进行概念类划分的基础之上,采用虚拟语言模型完成数据库的选择。该方法不需要数据库文档的统计信息或对文档进行训练,因此简单易于实现。对于搜索结果的合并,充分利用搜索结果隐含的信息,可有效减小搜索引擎之间的差异,试验结果表明查询效率明显高于 Web 搜索引擎。

参 考 文 献

[1] Wang W, Meng W, Yu C. Concept hierarchy based text database categorization in a metasearch engine environment// Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00). 2000:283-290

[2] Ponte J M, Croft W B. A language modeling approach to information retrieval// Proceedings of the ACM Conference on Special Interest Group on Information Retrieval. New York, N. Y., USA: ACM Press, 1998:275-281

[3] 张卫丰,徐宝文,周晓宇,等. 元搜索引擎结果生成技术研究. 小型微型计算机系统,2003,24(1):34-37

[4] Hoon G K, Tan S S, Tang E K, et al. Rank Aggregation model for meta search- an approach using text and rank analysis measures// Proceedings of the International Conference on Intelligent Information Processing (ICIIP 2004). London: Springer-Verlag, 2004:325-339

[5] Bordogna G. Soft fusion of information accesses// Proceedings of the 2002 IEEE International Conference on Fuzzy Systems. 2002 (2):1466-1471

[6] Wu S L, Mcclean S. Result merging methods in distributed information retrieval with overlapping databases. Information Retrieval, 2007, 10(3):297-319

[7] Keyhanipour A H, Moshiri B, Piroozmand M, et al. WebFusion: fundamentals and principals of a novel meta search engine// Proceedings of the 2006 International Joint Conference on Neural Networks. 2006:4126-4131

(上接第 80 页)

议和发展技术有机地整合起来,高健壮性地、高完备性地、高效率地、高准确度地快速发现网络拓扑信息。它不同于传统的面向协议的发现算法,是一种面向过程的算法,即,对发现过程分步骤,每一个阶段确定一个执行目标,是一种分级搜索的策略,将网络拓扑发现分成两级进行:一级拓扑发现主要发现路由器(网关,有路由功能的主机)设备和子网;二级拓扑发现主要发现子网内的主机以及子网类型等一些信息。该方法与网络的分层结构相一致,能更好地体现网络的层次。

不过移动代理技术刚兴起不久,发展还不够成熟,想要投入实际使用还有很多问题需要解决,例如管理者怎样动态影响移动代理的生命周期,或代理如何自主预测、避免和减少由于自身移动所可能出现的网络拥塞、系统崩溃、安全侵害等问题,研究更多的用于网络管理的移动代理,同时考虑更好的代理受控算法或代理安全策略,也是我们今后的研究方向。

参 考 文 献

[1] 杨家海,任宪坤,王沛瑜. 网络管理与实现技术[M]. 北京:清华大学出版社,2000

[2] 王志刚,王汝传,王绍棣,等. 网络拓扑发现算法的研究[J]. 通信学报,2004,25(8)

[3] 岑贤道,安常青. 网络管理协议与应用开发[M]. 北京:清华大学出版社,2002

[4] Dah M C, Ram S, Chiu D M. Network Monitoring Explained: Design And Application [M]. Massachusetts: Prentice Hall, 1992:50-180

[5] Ramadas S. Sppial Edition Using TCP/IP[M]. 北京:电子工业出版社,2003