

文章编号:1673-9469(2012)03-0103-06

一种基于奖励机制的智能元搜索引擎

黄伟建,祝月红,杜巍

(河北工程大学 信息与电气工程学院,河北 邯郸 056038)

摘要:针对元搜索引擎中返回大量重复冗余信息导致结果显示代理负担加大、系统查准率降低的缺陷,结合 Agent 技术建立基于多 Agent 的元搜索引擎系统模型,从成员 Agent 的爬行能力值、检索文档与查询主题的相关度和查询响应时间三个方面综合衡量成员搜索引擎对于查询的重要度,并按降序排序,优先选择重要度最佳的若干成员搜索引擎进行智能调度和智能结果合成。实验结果表明,与传统元搜索引擎相比,这种基于奖励机制的智能元搜索引擎提高了检索效率和查询性能。

关键词:奖励机制;元搜索引擎;智能代理;成员引擎调度策略;结果合成策略

中图分类号:TP391

文献标识码:A

An intelligent meta search engine based on reward mechanism

HUANG Wei-jian, ZHU Yue-hong, DU Wei

(Dept of Information Science & Electrical Engineering, Hebei University of Engineering, Hebei Handan 056038, China)

Abstract: To settle the defects that the repetitive and redundant information returned by meta search engine leads to increase the burden of result display agent and reduce the precision, a meta search engine system model based on multi-agent is brought forward. The importance degree of member search engines to a particular query comprehensively is measured, in the aspects of member Agent's crawling capability, the relevance between retrieve documents and certain query, and response time during query. Then the importance degrees of member search engines are ordered by descend, so as to select several member search engines with higher importance degree to dispatch and merge. It has proved through the experimental comparison that compared to the traditional meta search engine, the intelligent meta search engine based on reward mechanism improves the retrieval efficiency and query performance to a certain extent.

Key words: reward mechanism; meta search engine; agent; member search engine scheduling strategy; results merging strategy

在网络信息量呈指数增长的信息时代,任何一个独立搜索引擎的数据库容量都不足整个网络信息量的三分之一。由于信息覆盖面的限制,传统搜索引擎无法很好满足用户对查全率的要求。元搜索引擎的出现在一定程度上解决了这个问题。元搜索引擎将用户的查询请求发送到多个独立搜索引擎上同时进行检索,最后将查询结果统一整理后显示给用户^[1]。其工作原理如图 1

所示。

由于元搜索引擎的独特优势,人们对元搜索引擎的研究十分活跃。目前,国外主要的元搜索引擎有 Dogpile、Mamma、Profusion、Inquick 等,国内主要有万纬、MetaFisher 等比较典型的中文元搜索引擎。近几年,随着元搜索引擎技术的深入研究,相继出现了一些新的元搜索引擎模型,比如钟智等^[2]设计了一种智能化提供个性化服务的元搜索

提出一种新的基于奖励机制的成员搜索引擎调度策略。该策略的基本思想是,通过输入查询,发现对新查询潜在有用的成员搜索引擎,通过计算各成员搜索引擎对于查询的重要度,将重要度排名靠前的几个成员搜索引擎优先调度使用,并可通过多次查询,实时、动态调整调度的成员搜索引擎,使得每次查询都选择检索性能最佳的成员搜索引擎、以最优的方式完成检索。

2.2 具体步骤

基于奖励机制的成员搜索引擎调度策略的具体步骤如下:

步骤1:创建成员搜索引擎列表并生成对应的成员 Agent。查询前首先将尽量多的独立搜索引擎加入引擎知识库的搜索引擎列表中,生成与之对应的成员搜索引擎 Agent,并建立关联。

步骤2:初始化各 Agent 及其成员搜索引擎 Agent 的重要度。成员搜索引擎的重要度主要受三个性能因子影响:成员 Agent 的能力值、检索文档与查询主题的相关度以及成员搜索引擎查询的平均响应时间。各成员搜索引擎 Agent 的信息在成员引擎知识库中进行注册,初始化各成员搜索引擎 Agent 的重要度,并为其分配相同的初始值。

这里,定义 $C = \{C_1, C_2, C_3, \dots, C_n\}$ 为成员搜索引擎 Agent 的集合(n 为成员搜索引擎 Agent 的个数),用一个三元组表示第 i 个成员搜索引擎的重要度 $\text{Weight}(C_i)$,即 $\text{Weight}(C_i) = \{\text{Capability}(C_i), \text{Relevance}(C_i), \text{Response-Time}(C_i)\}$,其中 $\text{Capability}(C_i)$ 表示成员 Agent C_i 的能力值, $\text{Relevance}(C_i)$ 表示成员 Agent C_i 的检索文档与查询主题的相关度, $\text{Response-time}(C_i)$ 表示成员 Agent C_i 查询的平均响应时间。在初次查询开始之前,所有成员 Agent 的重要度参数相同。

步骤3:根据奖励机制,计算各性能因子。查询管理 Agent 将查询任务同时发送给所有搜索引擎列表中的各 Agent,返回查询结果后根据其表现情况进行奖惩,并记录在奖励机制库中。对成员搜索引擎 Agent 各性能因子的奖励机制具体如下:

1)成员 Agent 的能力值 Capability 。每个成员 Agent 爬行前,查询管理 Agent 会为其分配一定的爬行时间,设 t_c 为成员 Agent 已用的爬行时间, t_m 为系统分配的总时间, t 为成员 Agent 剩余时间与系统分配总时间的比值,即 $t = (t_m - t_c)/t_m$ 。查询管理 Agent 还会为成员 Agent 分配用于存储待爬

链接的最大阈值(一般用可存储链接的个数表示),设 s_l 为待爬队列中的链接个数, s_m 为阈值, s 为剩余空间与最大阈值的比值,即 $s = (s_m - s_l)/s_m$ 。那么,根据 t 和 s ,则某成员 Agent C_i 的能力值定义^[8]如式1:

$$\text{CAPABILITY}(C_i) = \begin{cases} \lambda t + (1 - \lambda)s, & t \times s > 0 \\ 0, & t \times s \leq 0 \end{cases} \quad (1)$$

能力值由 t 和 s 共同确定,系数决定了 t 和 s 对能力值的重要度,其中 $\lambda \in [0, 1]$ 。 $\text{CAPABILITY}(C_i)$ 的值越大,说明该成员 Agent C_i 的能力越高。如果某成员 Agent 没有剩余空间(即 $t = 0$)或无剩余存储空间(即 $s \leq 0$),也就是当 $t \times s \leq 0$ 时,说明该成员 Agent 已没有继续执行爬行任务的能力。

2)检索文档与查询主题的相关度 Relevance 。假设用 $\text{DOC} = (D_1, D_2, D_k, \dots, D_m)$ 表示某成员 Agent 检索到的文档集合,该集合按网页文档下载的時刻顺次排列, D_m 为最后一篇下载的文档。利用向量空间模型,文档 D_k 可表达成 $D_k = (d_{k1}, d_{k2}, \dots, d_{ki}, d_{kn})$ 的形式, d_{ki} 表示向量 D_k 的第 i 维。元搜索引擎的查询主题可表示为 $KW = (kw_1, kw_2, \dots, kw_i, kw_n)$ 的形式, kw_i 表示向量 KW 的第 i 维。那么在第 j 次查询中,成员 Agent 的检索文档与查询主题之间的相关度为

$$\text{relevance}(KW, D_k)_j = \frac{\sum_{i=1}^n KW_i \times d_{ki}}{\sqrt{\sum_{i=1}^n (KW_i)^2 \sum_{i=1}^n (d_{ki})^2}} \quad (2)$$

根据该成员 Agent C_i 近 n 次的执行情况设置一个阈值 $\text{Rel}(C_i)$,作为衡量标准。阈值的计算方法为

$$\text{Rel}(C_i) = \frac{1}{n} \sum_{j=1}^n \text{relevance}(KW, D_k) \quad (3)$$

然后比较 $\text{relevance}(KW, D_k)_i$ 和 $\text{Rel}(C_i)$ 的大小并观察 $\text{relevance}(KW, D_k)$ 的变化情况。设 $P(\text{relevance}_i)$ 为某成员 Agent 第 j 次查询时检索文档与查询主题相关度的奖惩值,奖惩函数如下:

①若 $\text{relevance}(KW, D_k)_j \geq \text{Rel}(C_i)$,说明成员 Agent C_i 的任务执行情况较好,给予奖励。奖励值为:

$$P(\text{relevance}_j) = \frac{(\text{relevance}(KW, D_k) - \text{Rel}(C_i))^2}{\text{Rel}(C_i)^2};$$

②若 $\text{relevance}(KW, D_k)_j < \text{Rel}(C_i)$,说明成员 Agent C_i 的执行情况差,给与惩罚。惩罚值为:

$$P(\text{relevance}_j) = \frac{(\text{relevance}(KW, D_k) - \text{Rel}(C_i))^2}{\text{Rel}(C_i)^2};$$

③若 $\text{relevance}(KW, D_k)$ 整体走势由低变高,超过了 $\text{Rel}(C_i)$, 说明该成员 Agent C_i 有较好的前景,可以予以奖励。奖励值为: $P(\text{relevance}_i) = 1$;

④若 $\text{relevance}(KW, D_k)$ 整体走势由高变低,低于了 $\text{Rel}(C_i)$, 说明该成员 Agent C_i 前景较差,给予惩罚。惩罚值为: $P(\text{relevance}_i) = -1$ 。

设用 $P(\text{relevance})_0$ 表示开始查询之前,检索文档与查询主题的相关度重要性的初始化值。那么,第 i 个成员 Agent 的检索文档与查询主题的相关度的重要性为:

$$P_{\text{rel}}(C_i) = P(\text{relevance})_0 + \sum_{j=1}^i P(\text{relevance}_j) \quad (4)$$

3) 查询响应时间 Response - Time。假设用 t_r 表示某成员 Agent 的近五次查询的平均响应时间, t_h 表示响应时间的阈值(默认值为 $15 \text{ s}^{[9]}$), t_o 表示可以被接受的最大响应时间(默认值为 45 s), $P_{\text{RT}}(C_i)$ 表示成员 Agent C_i 平均响应时间的重要性。

①若某成员 Agent C_i 的平均响应时间大于阈值,则对该成员 Agent C_i 进行惩罚,将其重要性减少:

$$\Delta P_{\text{RT}}(C_i) = -(t_r - t_h)^2 / (t_o - t_h)^2;$$

②若某成员 Agent C_i 的平均响应时间小于阈值,则对该成员 Agent C_i 进行奖励,将其重要性增加:

$$\Delta P_{\text{RT}}(C_i) = (t_h - t_r)^2 / (t_o - t_h)^2;$$

③若成员 Agent C_i 的平均响应时间恰好等于阈值,则不予惩罚和奖励,即 $\Delta P_{\text{RT}}(C_i) = 0$ 。

设用 $P_{\text{RT}}(C_i)_0$ 代表平均响应时间的初始化值,初始时各个成员 Agent 平均响应时间的重要性相同。因此,成员 Agent C_i 的平均响应时间的重要性可表示为:

$$P_{\text{RT}}(C_i) = P_{\text{RT}}(C_i)_0 + \sum \Delta P_{\text{RT}}(C_i) \quad (5)$$

步骤4: 计算各成员搜索引擎 Agent 的重要度。得到成员搜索引擎 Agent 各性能因子的评价值后,再加权综合评价即可得到成员 Agent C_i 的重要度,计算公式为:

$$\text{Weight}(C_i) = \alpha P_{\text{rel}}(C_i) + \beta \text{Capability}(C_i) + \gamma P_{\text{RT}}(C_i) \quad (6)$$

式中: α - 性能因子 $P_{\text{rel}}(C_i)$ 的权值; β - 性能因子 $\text{Capability}(C_i)$ 的权值; γ - 性能因子 $P_{\text{RT}}(C_i)$ 的权值。

性能因子的对于查询的影响越大,相应的权值就越高。

步骤5: 根据重要度排名,选择检索性能最佳

的若干(具体数目可根据个人需求设定)成员搜索引擎 Agent 进行查询调度。

成员搜索引擎调度策略是元搜索引擎的技术核心,通过选择良好的成员搜索引擎调度策略可以提高元搜索引擎的系统性能。

3 基于奖励机制的结果合成策略

结果合成策略是元搜索引擎的关键技术之一,结果合成策略的好坏在一定程度上决定着元搜索引擎性能的优劣。基于奖励机制的结果合成策略,克服了传统合成策略在检索结果合成排名时只考虑检索结果与查询主题相关度的缺陷,而是结合成员搜索引擎的重要度,将查询结果被各成员搜索引擎命中的次数也考虑在内,即由成员 Agent 的爬行能力值、检索结果与查询主题的相关度、成员搜索引擎的查询平均响应时间和查询结果命中次数四个影响因素来综合决定结果合成。

基于奖励机制的结果合成策略的基本思想是,通过计算某一查询结果在与其对应检索的成员搜索引擎中的局部重要性和在所有查询结果中命中数量的全局重要性,来综合评价该查询结果的整体重要性。基于奖励机制的结果合成策略的基本步骤如下:

步骤1: 计算查询结果在成员搜索引擎中的局部重要性等级值 $\text{WeightRank}(\text{Item}_i)$

首先按照6式计算成员搜索引擎对于查询的重要度 $\text{Weight}(C_i)$, 并计算成员搜索引擎的重要度等级值 $\text{WeightRank}(C_i)$ 为:

$$\text{WeightRank}(C_i) = \text{Weight}(C_i) / \sum \text{Weight}(C_i)$$

由于成员搜索引擎按其重要度降序排列,优先选择排名靠前的、检索性能最佳的若干成员搜索引擎进行调度,那么该搜索引擎的检索结果与查询主题的相关度越大。可认为查询结果在成员搜索引擎中的重要性等级值与该成员搜索引擎的重要性等级值呈正相关,即存在函数映射:

$$f: \text{WeightRank}(C_i) \rightarrow \text{WeightRank}(\text{Item}_i)$$

$$\text{且 } \text{WeightRank}(\text{Item}_i) \propto \text{WeightRank}(C_i) \quad (7)$$

步骤2: 计算查询结果命中数量的全局重要性等级值 $\text{VoteRank}(\text{Item}_i)$

在元搜索引擎中,由于不同搜索引擎可能使用相同的索引数据库,有时多个搜索引擎可能会出现相同的搜索结果。如果某个搜索结果 Item_j 被成员搜索引擎命中的次数越多,其符合查询主题的概率就越大。因此命中次数多的搜索结果去

重后可优先显示给用户。设搜索结果用集合 $Result = \{(Agent(C_i), Item_j) | 1 \leq i \leq n, 1 \leq j \leq m, \text{且 } n \in N^+, m \in N^+\}$ 表示, 搜索结果总数为 $ResultNum = \sum_{i=1, j=1}^{i=n, j=m} (Agent(C_i), Item_j)$, 成员搜索引擎 $Agent(C_i)$ 对某条搜索结果 $Item_j$ 命中的次数为 $VoteNum(Agent(C_i), Item_j)$, 计算其命中次数等级值 $VoteRank(Item_j)$, 计算公式为:

$$VoteRank((Item_j) = VoteNum(Agent(C_i), Item_j) / ResultNum \quad (8)$$

步骤3: 计算查询结果最终等级值 $Rank(Item_j)$

用 λ 和 φ 为影响因子系数。则由式7和式8, 可推导出查询结果 $Item_j$ 的最终等级值 $Rank(Item_j)$ 为:

$$Rank(Item_j) = \lambda \times WeightRank(Item_j) + \varphi \times VoteRank(Item_j) \quad (9)$$

式9中: $\lambda \in (0, 1), \varphi \in (0, 1)$, 且 $\lambda + \varphi = 1$ 。

4 性能验证

为验证基于奖励机制的智能元搜索引擎系统的性能和可行性, 将本系统模型在 JADE 平台上实现仿真和模拟, 并与元搜索引擎万纬的查全率、查准率和查询响应时间进行对比。由于万纬可以调用百度、搜狐、中文 Google、中文雅虎、新浪 GB、天网等 6 个中文搜索引擎, 以及 Google、Yahoo、Hot-Bot 等 3 个英文搜索引擎, 因此在本系统的成员引擎知识库中, 通过搜索引擎的 API 接口函数添加相同的 9 个中英文搜索引擎(没有提供 API 接口函数的搜索引擎可直接使用其检索结果), 并生成相应的成员搜索引擎 Agent。然后随机采取 20 个中英文关键词在本系统模型中搜索, 按照奖励机制对各成员搜索引擎 Agent 的重要度计算, 经统计分析, 得出重要度排名前 4 位的成员搜索引擎为: Google、百度、搜狐、Yahoo, 用以供本系统模型查询调度。在 20 次随机查询中, 本系统与万纬的查全率、查准率和查询响应时间情况的对比见图 3。

通过图 2 可以看出, 本系统采取奖励机制选择的查询性能最优的四个独立搜索引擎, 与万纬同时调用 9 个独立搜索引擎检索对比, 查全率几乎无太大影响, 查准率比万纬平均提高了 11.6%, 查询响应时间平均缩短了 2.3 s。

基于奖励机制的智能元搜索引擎与传统元搜索引擎相比, 有以下独特优势:

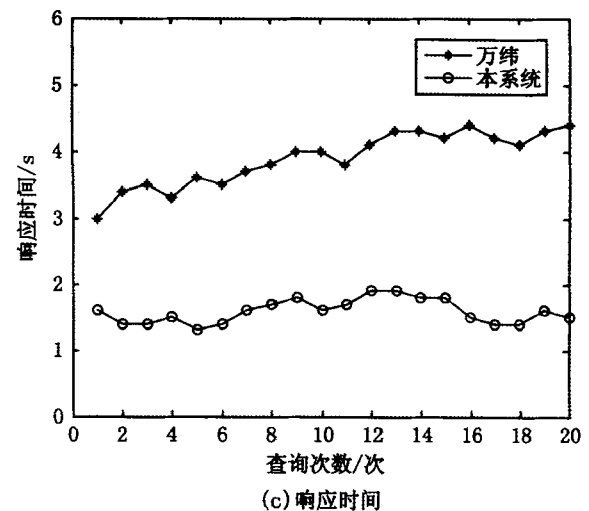
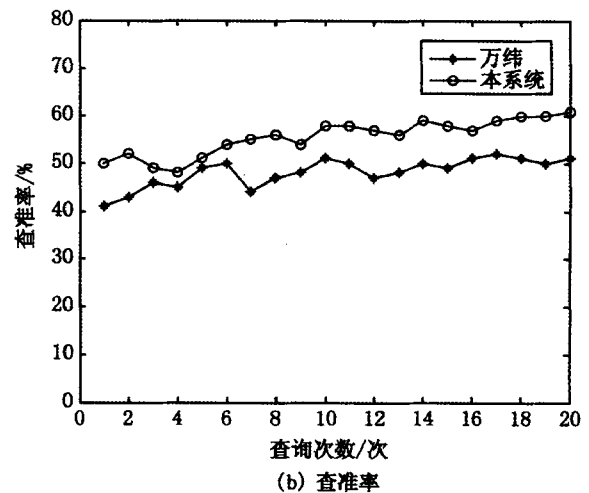
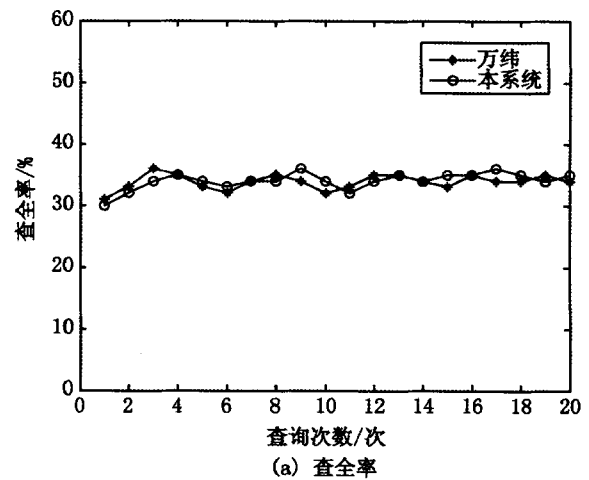


图3 查询结果对比

Fig.3 Result of comparison

(1) 每次查询都选择检索性能最佳的成员搜索引擎进行调度, 在保证查全率的基础上, 避免了无效成员搜索引擎检索返回的大量重复信息, 在一定程度上提高了查准率。

(2) 由于无效成员搜索引擎冗余信息的减少, 元搜索引擎后期结果处理的负担降低, 系统查询

响应时间也随之缩短。

(3)用定量法对成员搜索引擎的查询性能进行量化管理,通过完善的奖励机制发掘对查询潜在有用的成员搜索引擎进行调度,改进了传统成员搜索引擎调度策略的性能。

(4)将 *Agent* 技术与元搜索引擎结合,使系统更具灵活性和可扩展性。成员搜索引擎 *Agent* 建立后,可在成员搜索引擎知识库列表中按个人需求增删、修改对应的成员搜索引擎,满足不同用户的个性化需求。

5 结语

随着信息量的日益扩增,搜索引擎技术的发展称为人们备受关注的热点问题之一。本文结合前沿的 *Agent* 技术,基于奖励机制的智能元搜索引擎通过完善的奖励机制对成员搜索引擎进行调度,并根据奖励机制对查询结果进行归并,在提高查准率、缩短查询时间、减轻元搜索引擎后期的结果处理负担方面,都在一定程度上有所提高。系统模型中各影响因子系数的判定以及一些具体的实现技术,还需要进一步研究。

参考文献:

[1] 彭喜化. 基于 *Agent* 的元搜索引擎结果优化研究

[D]. 重庆:西南农业大学, 2004.

[2] 钟智,黄发良. 基于个性化服务的元搜索引擎模型[J]. 河北理工学院学报, 2005(01): 73 - 75.

[3] 黄国景,崔志明. 基于 *Ontology* 的个性化元搜索引擎研究[J]. 微电子学与计算机, 2004, 21(12): 100 - 103.

[4] 王小朋. 基于代理的元搜索引擎的研究[D]. 辽宁: 辽宁工程技术大学, 2005.

[5] 杨刚华. 基于 *Agent* 的个性化信息检索系统研究[D]. 大连: 大连理工大学, 2005.

[6] 李晓瑜. *Multi-Agent* 在网络海量信息搜索中的应用[J]. 科技信息, 2009(14): 92.

[7] ASHRI R, RAMCHURN S D, SABATER J, et al. Trust evaluation through relationship analysis[C]// Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. Utrecht, The Netherlands: AAMAS05, 2005: 1005 - 1011.

[8] 孟文杰. 元搜索引擎的调度策略研究[D]. 青岛: 中国石油大学, 2007.

[9] 黄伟建,宋晓洁,王子轩. 办公自动化系统中动态工作流引擎的研究[J]. 河北工程大学学报: 自然科学版, 2011, 28(4): 45 - 48.

[10] DREILLINGER D, HOWE A E. Experiences with selecting search engines using meta search[J]. ACM TOIS, 2007, 15(3): 195 - 222.

(责任编辑 马立)