中国科学: 信息科学 2015 年 第 45 卷 第 5 期: 605–622
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE CHINA PRESS

论 文

基于 Agent 的智能化元搜索引擎个性化机制

李青山[①②③*], 王俊[①③], 褚华[①②], 季陶然[①③]

① 西安电子科技大学软件工程研究所, 西安 710071
② 西安电子科技大学软件学院, 西安 710071
③ 西安电子科技大学计算机学院, 西安 710071
* 通信作者. E-mail: qshli@mail.xidian.edu.cn

收稿日期: 2014–04–10; 接受日期: 2014–08–06; 网络出版日期: 2014–12–16
国家自然科学基金 (批准号: 61373045, 61173026, 61202039)、中央高校科研业务基金 (批准号: K5051223008, BDY221411)、十二五国防预研项目 (批准号: 513***103E) 和国家高技术研究发展计划 (863 计划)(批准号: 2012AA02A603) 资助

摘要 大数据环境下, 信息量过载, 人们需要精准、智能的检索工具. 本文研究了基于 Agent 的智能元搜索引擎中的个性化机制, 准确地理解用户的搜索意图, 有效地提高了信息检索的服务质量. 文中着重研究基于 Agent 的智能元搜索引擎个性化方法及功能实现所需的相关理论与技术, 给出了查询语句分析与查询兴趣挖掘及成员搜索引擎调度过程, 设计了基于动态学习的复杂查询识别机制, 基于动态更新的用户兴趣概貌模型的检索兴趣挖掘机制, 以及基于概念格与日志分析的搜索引擎评估调度策略机制. 最后, 针对复杂查询语句识别、搜索引擎调度策略效果及检索结果相关性的测试结果表明, 本文提出的基于 Agent 的智能元搜索引擎个性化机制, 可较为准确地识别出复杂的查询语句并进行预处理, 高效学习用户的查询兴趣, 达到明显提高检索结果相关程度的目的, 并智能化地调度成员搜索引擎, 为提高用户信息检索效率提供充分支持, 从而提高用户的检索体验.

关键词 信息过载 元搜索引擎 个性化机制 Agent 检索相关程度

1 引言

大数据环境下, 人们需要精准、智能的检索工具. 根据美国市场研究公司 IDC 的研究报告, 全球的数据产生量在 2013 年已达到 2.7 万亿 GB, 而且未来十年全球数据存储量将增长 50 倍, 人们已身处大数据时代 [1]. 互联网是人们获取海量数据的主要渠道, 网页是数据的主要载体. 但是 Internet 上的信息通常以无组织的形式分布于开放、异构的节点中, 信息的可利用性和可靠性在不断地变化, 因此人们需要精准、智能的检索工具来应对 “大数据” 所带来的信息过载. 搜索引擎一直是人们查找与定位互联网数据信息的重要工具, 但面对信息资源不断膨胀及用户需求不断提高的现状, 传统搜索引擎逐渐暴露出以下两个方面的问题.

一是信息覆盖率低且不同搜索引擎的检索结果重合率低, 而用户的 “信息类” 查询占绝大多数, 导致单一搜索引擎难以满足需求. 二是智能化服务水平低, 个性化功能不足且缺乏兴趣的主动学习与信息推荐能力. 现有的搜索引擎提供了一定程度的智能化检索服务, 例如基于地理定位的检索、热点推荐、相关搜索提示等, 但仍不能满足不同背景、不同目的、不同兴趣用户的个性化检索请求 [2].

引用格式: 李青山, 王俊, 褚华, 等. 基于 Agent 的智能化元搜索引擎个性化机制. 中国科学: 信息科学, 2015, 45: 605–622, doi: 10.1360/N112014-00079

通过对用户检索兴趣的分析与建模, 使搜索引擎能够根据用户兴趣背景的不同对相同的查询请求返回符合用户各自意图的检索结果, 实现信息检索的个性化处理. 已有的相关研究主要思路是对用户的 web 行为数据, 包括历史搜索记录、历史浏览记录、网页收藏记录等进行挖掘分析以捕获用户的兴趣, 进而对检索结果进行个性化分析与过滤. 美国的 Gauch 和德国的 Pretschner 较早开始这方面的探索, 提出了利用本体对用户概貌进行建模, 在搜索时根据用户概貌中的用户兴趣信息和查询词共同决定网页的排序 [3], 但由于系统需要集成到浏览器中, 所以并不适合用户实际使用.

检索条件是用户在信息检索时对搜索引擎提出的不同形式的检索请求, 包含短语、短语组合、语句等形式. 检索条件的复杂与否是影响搜索引擎检索效果好坏的重要因素, 如果能在检索前对检索条件提前做出复杂性判别, 进而采取相应的处理机制, 将大大提高搜索引擎的检索效果. 关于复杂查询识别方面的研究, 一是对检索条件分类方法的研究. Metzler 等 [4] 通过统计的方法来实现问句的分类, 通过对真实的经过标注的问句语料进行统计学习, 提取能表达各种问句类型的特征规则, 建立学习模型, 实现各种问句的类型识别. 二是对汉语句型识别的研究. 较早的是清华大学罗振声等 [5] 提出的以谓语为中心的句型成分分析与句型匹配相结合的分析算法和策略. 三是对短语的识别. 如奚建清等 [6] 基于 HMM 的汉语介词短语自动识别.

目前伴随搜索引擎种类与功能的多样化发展, 搜索引擎评价研究逐渐兴起和发展. 2002 年, Wu 等 [7] 描述的统计模型表示为: 为每一个搜索引擎返回结果集建立两个指标, NoDoc 和 AvgSim, 表示结果集中高相关度的文章数目以及平均相似度. 但对于各个搜索引擎返回结果集的记录数变化比较大且需要进行快速处理的元搜索而言, 该模型是不理想的. 2005 年, Beg [8] 描述了基于用户行为的评价模型. 该方法通过对用户动作进行监听来实现对各个搜索引擎的评价, 但只适合对高级用户的动作进行评价. 由此可见, 国内外对搜索引擎的性能评价进行了一定的研究, 但大多没有给出详细的适用于元搜索引擎调度的成员搜索引擎评价体系, 也没有充分结合搜索引擎的搜索日志为用户提供个性化、高效的成员搜索引擎调度策略.

由以上分析可知, 用户检索个性化处理机制国内外已有较多研究, 但某些方面仍存在不足, 因此集成多个搜索引擎的搜索结果、提供统一访问机制并具有个性化的查询与推荐功能的元搜索引擎具有重要的研究意义. Agent 是分布式计算与人工智能结合的产物, 在一定环境中运行, 并能够根据环境的变化, 灵活、自主地采取行动以实现其设计目标的计算实体 [9], 且 Agent 具有感知性、适应性、自治性、主动性和协作性等特征 [10], 因而 Agent 成为构建智能化元搜索引擎的首选技术.

本文基于以上需求进行了元搜索引擎智能化技术的研究, 设计并开发了基于 Agent 的智能元搜索引擎系统. 基于 Agent 的智能元搜索引擎能够提供个性化的查询和推荐功能, 为检索用户提供个性化服务, 在大数据环境下提高查全率, 有效提高检索的相关性, 为用户带来更好的搜索体验. 同时, 本文提出的个性化方法与策略不仅适用于元搜索引擎, 也可作为传统搜索引擎改进查询效果的技术依据, 具有较高的应用价值. 智能化的元搜索引擎也具有广阔的应用前景. 通过扩大成员搜索引擎的领域属性范围, 智能化元搜索引擎可在多个应用领域发挥重要作用.

本文的后续部分组织结构如下: 第 2 节给出了基于 Agent 的智能元搜索引擎的体系结构. 第 3 ~ 5 节在总体框架指导下, 设计支持查询语句分析、查询兴趣挖掘、成员搜索引擎调度的基于 Agent 的智能元搜索引擎个性化模块, 包括复杂语句的识别方法、动态更新用户兴趣概貌模型以及基于概念格与日志方法的搜索引擎评估调度策略. 第 6 节采用典型的应用场景和具体的搜索用例, 进行了针对性的实验, 对本文提出的模型、策略和机制进行验证, 并进行了功能、性能等方面的分析. 第 7 节概括性地总结本文所做的工作, 并对下一步需要改进的研究方向和工作内容进行展望.

2 基于 Agent 的智能元搜索引擎体系结构

搜索个性化功能的实现要以支持该功能的系统体系结构为基础. 本节针对元搜索引擎智能化功能需求, 给出基于 Agent 的智能元搜索引擎的结构模型及交互关系, 以支持个性化模式的信息过滤、群组用户兴趣学习与推荐以及检索结果的有效合成.

为构造智能化的元搜索引擎, 本文基于 FIPA 标准并结合元搜索引擎个性化搜索、主动感知环境变化、查询主动推荐的实际需求, 设计了基于 Agent 的智能元搜索引擎结构模型. 该模型充分利用 Agent 自主性、协作性、感知性等固有特点, 使系统建模过程中具有较高的抽象性和自然性, 同时对有大量用户访问、运行环境动态多变的互联网环境具有较高的灵活性和适应性. 本文设计了三类 Agent 内部结构及数据交互关系. 其中推荐管理 Agent 通过分析用户在服务器端的检索信息, 挖掘用户的检索兴趣, 然后通过 Agent 间的相互协作获得群组推荐查询和推荐检索结果. 检索合成 Agent 接收用户的查询请求, 进行对复杂条件进行转换, 对查询进行分类, 然后根据调度策略调用成员搜索引擎并对检索结果及相关推荐信息进行合成处理, 同时可感知成员搜索引擎的状态变化. 调度管理 Agent 根据查询请求选择符合用户需要的成员搜索引擎, 产生调度策略以及协作学习搜索引擎搜索能力. 三类 Agent 内部结构及数据交互关系如图 1 所示. 本文重点研究的是用户个性化查询功能, 该功能主要涉及三类 Agent, 包括推荐管理 Agent、调度管理 Agent 及检索合成 Agent.

3 基于动态学习的复杂查询语句识别机制

用户在搜索引擎中输入查询后并不总是能够找到需要的信息, 其原因除了用户所需的信息在互联网上根本不存在或是存在但没有被搜索引擎收录外, 更多的则是查询语句表达方式的问题. 因为搜索引擎虽然收录了用户需要的某些信息, 但用户的查询若不能很好地与目标信息相匹配, 则检索出的结果便容易偏离用户的意愿. 比如查询为 “股票现在应该买还是应该卖”、“手里股票可以不必急于卖掉” 等信息便不容易被检索出来.

出现上述问题的主要原因是目前搜索引擎采用的是基于关键字匹配的检索方式, 无法理解检索条件的语义, 若查询里的关键字没有或较少在某条信息中出现, 则该信息会被搜索引擎判定为不相关内容, 用户便难以较快找到该信息. 针对这一问题, 本文提出了基于动态学习机制的复杂查询语句识别方法, 用于识别和简化复杂查询语句, 目的在于在现有搜索引擎工作模式的背景下, 尽可能地提高查询语句与检索结果内容之间的匹配程度, 实现较好的检索效果.

因此, 本文的 “个性化” 第一方面表现是对查询语句进行分析, 识别出 “复杂” 的查询语句并提示用户可重新构造 “简单” 查询语句, 使其更容易被搜索引擎处理, 得到更相关的检索结果. 整个过程由检索合成 Agent 完成, 首先对用户查询语句进行复杂性分析, 如果用户查询是非复杂查询语句, 则不进行提示, 若用户查询是复杂查询语句, 则检索合成 Agent 在检索结果页面对用户做出提示.

3.1 复杂查询语句相关定义

本文针对搜索引擎的实际使用需求, 将查询语句分为两类: 简单查询语句和复杂查询语句. 简单查询语句所涉及的语句形式种类繁多, 包括短语 (phrase, PHR) 或短语组合 (phrase combination, PHC) 及表达意图较为简单的句子 (simple sentence, SIS), 经分词后进行检索可以达到较好的效果.

而复杂查询语句主要指语义结构复杂或限定条件较多的语句, 比如 “冲洗鼻咽的洗鼻器如何用”、“为什么国外的互联网公司难以进入中国市场” 等, 这种类型的查询语句难以通过简单的分词后检索

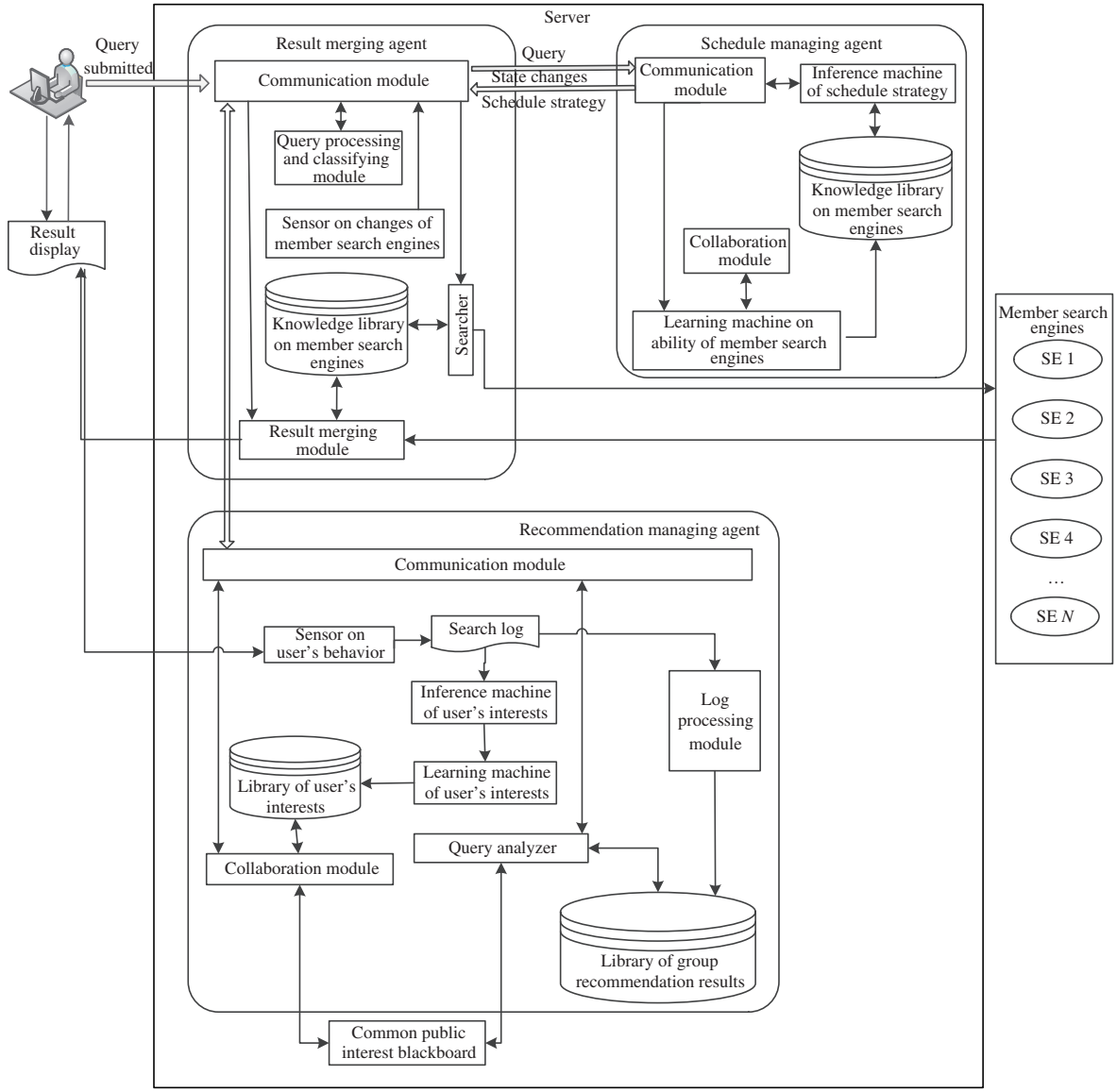


图 1　系统 Agent 结构模型及交互关系

Figure 1　Structure model and interaction of the system agent

到相关的结果, 因为包含相关结果的资源可能采取了另一种表述方式, 并不完全包含用户查询语句中的关键词. 对于复杂查询语句, 需要进行识别以便检索前预处理. 而简单查询语句则可直接进行检索. 其相关定义如下.

定义 1　查询语句复杂度 (retrieve query complexity, RQC). 对查询语句进行分词后, 去掉无核心语义功能的如 “的、了” 等助词及 “哼、啊” 等叹词后剩余词的数量 N.

定义 2　复杂语义结构 (sophisticated semantic structure, SSS). 初始的结构定义为含有集合 S 如 {“V+Dno/Dnot/Dornot+V”、“A+ Dno/Dnot/Dornot +A”、“VV”、“AA”}中的一个或多个元素. 其中 V 指动词, A 指形容词, Dno 指副词 “不”, Dnot 指副词 “没”, Dornot 指副词 “还是不”, “+” 表连接关系, 每个结构中的符号指代同一个词. 通过动态学习后会增加新的复杂语义结构.

定义 3 复杂查询语句 (sophisticated retrieval condition, SORC). 含有集合 S (RQC 超过 L, 出现 SSS) 中一个或多个元素的查询语句为复杂查询语句.

查询语句复杂度和复杂语义结构都是判断某个查询语句是否为复杂查询语句的标志. 本文从复杂查询语句入手, 确定其评价标准并以此判定某查询语句是否为复杂查询语句, 若经分析后不是复杂查询语句则可将其归为简单查询语句.

3.2 基于动态学习的查询语句复杂性分析

对查询语句的分析需要完成中文分词、词性标注、查询类型判别三个步骤. 其中中文分词及词性标注使用中国科学院的中文分词处理软件工具包 3.0 版. 它不但可以完成中文分词功能, 还可以给出词性标注, 分词准确率高达 97.58% (973 专家组评测). 同时它可以允许自定义分词及词性标注规则, 可以较好地完成专有名词、特殊习语的分词及词性标注, 本文将副词 “不” 自定义为 Dno, 副词 “没” 自定义为 Dnot, 副词 “还是不” 自定义为 Dornot.

在对查询语句 query 完成中文分词和词性标注后, 进行查询类型的判别. 首先计算条件的 RQC 值, 在计算 RQC 值之前应去掉词性为助词及叹词的词, 将 RQC 值记为 N, 将分词结果记为 query_div. 然后采用式 (1) 计算查询语句与复杂语义结构 SSS 的相似度.

$$\mathrm{sim}\,(\mathrm{query\_div}) = \frac{\sum_{\mathrm{term}_j \in \mathrm{query\_div}} \left(\sum_{s_i \in \mathrm{SSS}} \mathrm{same}(s_i, \mathrm{term}_j)\right)}{|\sum_{\mathrm{term}_j} (\mathrm{term}_j)| * |\sum_{s_i \in \mathrm{SSS}} (s_i)|}, \tag{1}$$

其中, $\mathrm{same}\,(s_i, \mathrm{term}_j) = \frac{\sum_{k=1}^{n} (tw_{ik} * tw_{jk})}{\sqrt{\sum_{k=1}^{n} tw_{ik}^2} * \sqrt{\sum_{k=1}^{n} tw_{jk}^2}}$. s_i 为 SSS 集合中的元素, term_j 为 query_div 中的元素. $\mathrm{same}(s_i, \mathrm{term}_j)$ 为 s_i 和 term_j 的相似度, 如果 s_i 和 term_j 完全相同则 $\mathrm{same}(s_i, \mathrm{term}_j)$ 值为 1. 该公式利用向量之间夹角余弦求其值, tw_{ik} 为元素 s_i 中第 k 个字的权重, n 为 s_i 中字向量空间的大小. 在集合 SSS 中, 用户对不同的元素的使用频率并不相同, 因此需要计算不同元素的权重, 在计算 RQC 值时应按照权重从高到低进行匹配计算. 所用公式如式 (2) 所示.

$$F_{\mathrm{SSS}}\,(s_i) = \frac{n(s_i)}{\sum_{s_i \in \mathrm{SSS}} n(s_i)}, \tag{2}$$

其中 $n(s_i)$ 指元素 s_i 到目前为止被用户使用的次数, 为所有元素的被使用次数之和. $f(s_i)$ 为 s_i 的被使用频率.

sim(query_div) 的值需要有限定范围, 规定查询语句与复杂语义结构的最低相似度为 query_limit, 如果查询语句 query_div 中出现了 SSS 集合中的某个元素, 则 sim(query_div) > query_limit, 否则 sim(query_div) < query_limit.

同时设定元素 s_i 与 term_j 的最低相似度 term_limit 值, 记录 same 值那些高于 term_limit 但低于 1 且不属于 SSS 集合的查询语句结构. 并增加该结构的 Ftemp 频率值. 然后分析考察其是否能够被搜索引擎较好地检索, 对于那些检索相关度较低的结构, 动态加入 SSS 集合中. Ftemp 值计算公式为 $F\mathrm{temp}\,(y) = \frac{n(y)}{N}$. 其中 y 为 query_div 的一个子结构, $n(y)$ 是一定时间内 y 的用户使用次数, N 为一定时间内用户的检索次数.

由于汉语句型的复杂性及语言的不断发展, SSS 集合中没有的元素也可能成为用户大量使用但检索效果不好的新的复杂语义结构, 因此有必要进行动态的学习以不断更新 SSS 集合中的元素. 本文设计了复杂性语义结构的动态学习流程, 在查询语句复杂性分析的过程中可记录新产生的复杂语义结构.

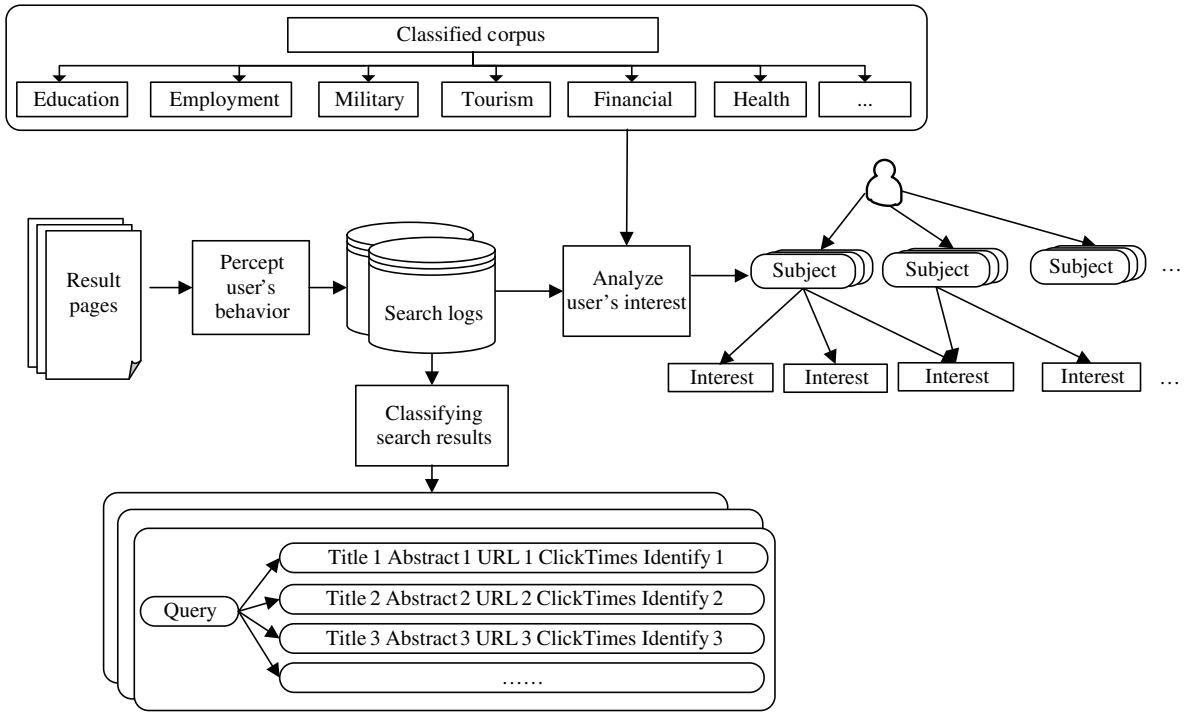


图 2　用户检索兴趣挖掘过程

Figure 2　Mining process of user query interest

在根据上文所述公式和复杂语义结构集合进行查询语句的复杂性分析后, 若有新的复杂语义结构, 则记录该新结构. 然后计算新复杂语义结构出现的频率, 分析新结构的检索效果. 接下来增加检索效果较差的高频新复杂语义结构到复杂语义结构集合中, 作为后续查询语句复杂性分析的依据. 整个过程为动态循环过程, 通过以上步骤完成复杂语义结构的动态学习. 由于篇幅有限, 检索条件复杂性分析算法详情请见附录 A.

在识别出复杂查询语句后, 由于语言本身的多义性和复杂性问题, 所以本文并未直接将其转换为简单查询语句, 而是给出用户提示. 系统将提示用户该条查询语句结构较为复杂或长度略长, 可能会影响检索结果精度, 建议用户重置查询以获得更好的检索效果, 从而可为实现个性化搜索功能提供支持. 关于复杂查询语句可否有效转化为简单查询语句的问题本文将在后续工作中进一步予以研究.

4　基于动态更新的用户兴趣概貌模型的检索兴趣挖掘机制

4.1　检索兴趣挖掘过程

个性化功能的第二方面表现是对用户检索兴趣进行挖掘, 从而对原始的搜索引擎检索结果进行过滤, 选择出符合用户检索意愿的个性化检索结果. 用户兴趣模型是实现智能化、个性化服务的前提条件. 将用户兴趣模型应用到检索结果过滤、查询信息推荐等功能中即可实现检索的个性化. 本文提出一种基于用户检索行为信息、在多方位协同分析用户兴趣偏好的检索兴趣挖掘方法, 进而创建用户兴趣模型, 更加准确和可靠地表达出用户检索兴趣. 其过程如图 2 所示.

用户查询兴趣的挖掘与应用由推荐管理 Agent 完成. 在服务器端, 推荐管理 Agent 感知用户在检

索结果页面的点击与浏览行为, 记录用户查询数据并生成用户查询日志. 推荐管理 Agent 一方面根据分类语料库对用户日志中查询与点击记录进行分析, 并创建能够反映用户检索兴趣层次的兴趣树作为用户兴趣模型, 并能主动感知与更新用户兴趣变化; 另一方面对用户查询与点击页面进行归类、去噪, 形成包括标题、摘要、URL 等结构化的查询结果数据, 其可作为群组推荐查询和推荐检索结果的原始数据. 在客户端, 兴趣挖掘 Agent 在移动 Agent 运行环境中运行, 在用户允许的情况下挖掘用户的浏览历史, 并将相关数据交由推荐管理 Agent 进行分析. 本文不探究兴趣挖掘 Agent 的设计问题. 依靠用户兴趣可以实现定制化的个性化功能, 如特定主题的结果展示与相关信息推荐等, 有效提高检索相关性, 提升用户检索体验.

4.2 用户兴趣概貌模型的设计与动态更新

用户兴趣模型是个性化服务的基础与核心, 是用来表示用户兴趣特征、描述用户喜好的模型, 描述了用户个人的兴趣信息. 而用户兴趣建模则是通过用户主动参与或者被动观察, 对其相关行为动作进行挖掘, 最终得到用户兴趣模型的过程. 本文采用系统主动学习的方法进行用户兴趣获取, 能减少用户显式参与, 降低复杂度, 而且客观上提高了用户兴趣模型的精确度.

本文提出了一种可以有效反映用户检索兴趣层次的用户 — 主题 — 兴趣三层的用户兴趣树作为用户兴趣模型, 如图 2 中所示. 通过上一节中对用户浏览历史记录的分析, 针对每个用户, 将分析得到的若干用户兴趣根据主题进行划分并组成一个三层的兴趣树, 并在用户使用元搜索引擎过程中持续感知记录其浏览行为以做对其兴趣的追踪分析, 并更新用户兴趣模型.

本文在研究了典型的兴趣模型后, 设计了一种全新的能进行遗忘学习、更新和优化, 较好地表达以及动态更新用户的兴趣特征的用户概貌. 本文设计的用户概貌由多个兴趣特征向量组成, 每个用户兴趣向量由一个五元组来表示, 即 $\{k_i, c_i, d_i, w_i, r_i\}$.

其中, k_i 表示词条, 即用户的兴趣词; c_i 表示分类, 即用户兴趣词的归属类别; d_i 表示更新日期, 即该词条最后一次更新的日期, 主要是为了便于遗忘因子根据日期对权值进行处理和进行权值的统一比较; w_i 表示权值, 即该词条在其所显示的更新日期时所具有的权值; r_i 表示关联词, 是指通过数据挖掘得到的, 与兴趣词 c_i 可能相关的若干词条, r_i 不是必须存在的.

用户兴趣词数量是有限的, 这是为了更好的将用户现阶段感兴趣的词表现出来, 文献 [2] 中指出用户兴趣特征词数量取 $40 \sim 60$ 之间比较合适, 在此将用户兴趣词数量设为 40 个. 兴趣词分类由所在文档的类别决定, 这样进行分类的好处就是对一词多义现象有较好的区分, 比如关键词 “苹果”, 可以是美国苹果公司及其产品, 属于 IT 类; 也可以是水果苹果, 属于健康类等.

本文在经过拉普拉斯 (Laplace) 平滑后的朴素贝叶斯 (Bayes) 方法的基础上进行基于词频的改进, 来对文档进行分类. 这里选用了搜狗实验室的文本分类语料库, 其中的十个分类分别是: IT、财经、健康、体育、旅游、教育、招聘、文化、军事和汽车. 具体的分类方法见附录 B.

兴趣特征向量中包含的通过关联规则挖掘得到的与兴趣词相关的若干关联词, 其可以从侧面描述兴趣词的性质, 甚至可以作为查询优化推荐给用户. 本文根据实际情况将分别将关联规则挖掘的最小支持度和最小置信度设定为 0.4 和 0.8. 为了不影响系统性能, 目前仅要求按照关联规则最多扩展出三个关联词即可. 在使用搜索引擎时, 使用多个查询词往往能更好的表达查询意图, 亦能获得更贴近用户需求的检索结果, 比如查询 “西电” 时如将其扩展为 “西电大学” 能有效过滤掉与 “大学” 无关的检索结果, 更可以直接返回与 “西安电子科技大学” 主题相关的结果.

因为用户兴趣模型空间大小有限, 故应对用户模型中关键词进行调整, 以适应用户兴趣随时间变化而变化, 并且保证用户当前时期最关注的兴趣词的权重值保持最高. 所以在这里引入了遗忘因

子 [11], 遗忘公式如式 (3):

$$F(k_i) = \mathrm{e}^{-\frac{\log 2}{hl}(\mathrm{now}-\mathrm{created})}. \tag{3}$$

其中 now 表示当前日期, created 表示兴趣词在兴趣模型中上次被更新的日期, hl 表示半衰期, 即经过 hl 天后用户的兴趣遗忘一半, 我们这里选用了 7 天作为其半衰期. 通过引入对特征值的遗忘算法, 可以有效的实现对用户兴趣动态变化的及时更新.

在对用户兴趣模型更新过程中, 首先获取到用户兴趣特征向量 new_vector $= \{\mathrm{word}_i, c_i, d_i, w_i^c, r_i\}$, 其中 word_i 为用户兴趣词, c_i 为兴趣词所属类别, d_i 为兴趣词更新时间, w_i^c 为兴趣词的权重值, r_i 为与兴趣词相关的关联词. 对用户兴趣模型中的所有兴趣特征向量进行遗忘计算, 利用遗忘因子 $F(k_i)$ 调整每个兴趣词的权重值; 对每个兴趣特征向量中的兴趣词权重值 w_i, 将其权重值更新为 $F(k_i) \times w_i$, 并将兴趣特征向量中兴趣词更新时间更新为执行操作的时间. 然后判断已存在的用户兴趣特征向量中是否存在与 new_vector 中主题类别相同、关键词一致的兴趣特征向量. 如果存在则计算刚才找到的兴趣特征向量的兴趣词权值 $w_i = w_i^{'} + w_i^c$, $w_i{}^{'}$ 是经过前面遗忘计算后的兴趣词的权值, w_i 为其新权重值; 将兴趣特征向量中兴趣词更新时间更新为执行操作的时间, 用户兴趣更新则结束; 如果不存在, 则判断兴趣特征向量数目是否已达到或者超过阈值, 如果达到或超过, 找出具有最小兴趣词权重值的兴趣特征向量, 将其在用户兴趣模型中删除, 否则将获取到的用户兴趣特征向量 new_vector 加入到用户兴趣模型中, 用户兴趣更新完成. 鉴于篇幅, 具体算法步骤见附录 C.

本文在研究了典型的兴趣模型后, 设计了上述全新的能进行遗忘学习、更新和优化, 较好地表达以及动态更新用户兴趣特征的用户概貌. 为实现检索个性化提供技术支撑, 有助于提高检索结果的相关性.

5 基于概念格与日志方法的成员搜索引擎评估的调度策略机制

基于 Agent 的智能化元搜索引擎的调度策略是指调度管理 Agent 根据查询请求, 依据成员搜索引擎搜索能力与用户偏好, 选择符合用户需求的成员搜索引擎的方法. 目的在于为用户选择数量合适并贴近用户查询需求的成员搜索引擎, 以较小的资源耗费, 帮助用户获得较高的查询质量与更符合用户检索意愿的个性化检索结果. 传统搜索引擎调度策略并不具有动态更新能力, 本文设计的该管理模块可以满足动态调整调度策略的需要, 具备较好的适应性. 本文结合成员搜索引擎搜索擅长能力与用户偏好两类因素构造搜索引擎搜索能力模型, 提出了基于概念格和日志的搜索引擎评价方法, 并结合用户兴趣模型设计实现了可符合用户偏好的成员搜索引擎调度策略. 这是本文个性化功能的第三个方面的体现.

5.1 搜索引擎能力模型

调度管理 Agent 根据查询请求产生成员搜索引擎调度策略, 其调度策略制定的主要依据是成员搜索引擎的搜索能力. 搜索引擎调度策略以搜索引擎的搜索能力为主要依据, 本项目对搜索引擎能力建立二元组模型 SEA $= \langle \mathrm{UserPreference}, \mathrm{StrongTheme} \rangle$, 其中 UserPreference 表示用户对搜索引擎的偏好程度; StrongTheme 表示该搜索引擎所擅长的相关主题概念. 用户使用搜索引擎过程中, 会发现某些搜索引擎具有更好的检索效果而产生个人偏好, 因此在搜索引擎的选择过程中应考虑到用户对搜索引擎的偏好程度; 另外各搜索引擎对不同主题概念的检索能力也有差异, 反映了该搜索引擎所擅长的相关主题概念. 本文结合以上两类因素构造搜索引擎搜索能力模型.

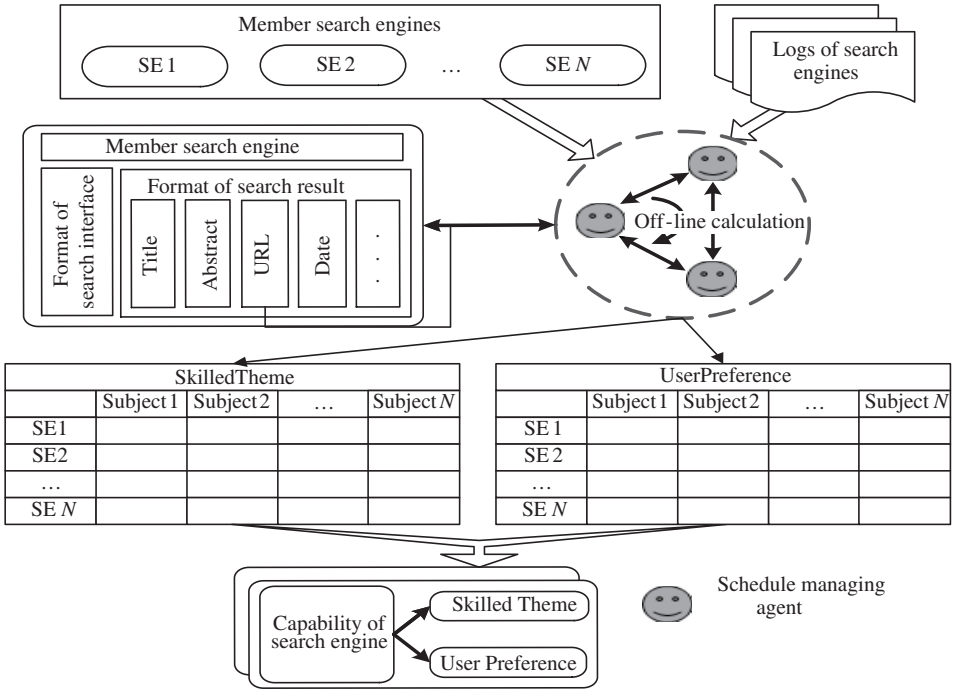


图 3 搜索引擎能力模型构造过程

Figure 3 Construction process of search engine capacity

调度管理 Agent 中搜索引擎能力学习机以离线计算的方式构造搜索引擎能力模型, 其可以依据不同成员搜索引擎的检索接口规则发起检索请求, 然后根据返回的检索结果, 构造概念格计算成员搜索引擎相似度, 通过发起查询多个属于不同主题的词, 最终可得到关于搜索引擎对不同主题擅长度的搜索引擎能力评价矩阵, 即得到 StrongTheme 元素. 在使用成员搜索引擎过程中, 如果搜索引擎的检索格式如接口、结果格式等发生改变, 则通知系统管理员修改相应格式. 另一方面调度管理 Agent 也可以通过对搜索引擎日志进行分析, 发现用户使用搜索引擎的喜好, 计算得到基于用户偏好的搜索引擎能力评价矩阵, 即得到 UserPreference 元素. 最终完成搜索引擎能力模型的构造, 相关过程如图 3 所示.

计算得到搜索引擎能力模型后, 调度 Agent 在用户检索时结合用户兴趣模型生成调度策略. 用户进行检索时, 调度策略推理机首先结合用户兴趣模型分析检索词所属的主题分类, 根据离线计算得到的搜索引擎能力模型, 计算得出最擅长该主题并符合用户偏好的若干成员搜索引擎, 系统就可以根据该结果调度成员搜索引擎.

由此可以看出, 生成调度策略的核心技术之一就是构造搜索引擎能力模型, 而构造搜索引擎能力模型的核心就是搜索引擎评价, 即对搜索引擎检索结果的质量评价. 元搜索使用自动化检索结果评价可以快速准确的选择适合当前查询请求的成员搜索引擎提供检索结果. 接下来两节分别介绍基于概念格与基于日志分析的搜索引擎评价方法.

5.2 基于概念格与日志的搜索引擎评价调度算法

5.2.1 基于概念格的搜索引擎评价方法

本文引入形式概念分析, 提出了基于概念格的搜索引擎评价方法, 来衡量成员搜索引擎的搜索能

力, 为制定调度策略提供依据. 形式概念分析是一种从形式背景中建立概念格来进行数据分析和规则提取的强有力工具, 它的相关数据结构是概念格, 相关具体介绍见附录 D. 构造概念格时, 需要建立形式背景. 概念格的形式背景为三元组 (O,D,R), 分别表示对象、属性以及对象与属性之间的关系. 在搜索结果集中, 提取网页的 URL 作为形式背景中的对象集合 (URL), 将网页标题以及摘要信息经过分词处理后作为形式背景的属性集合 (keyword), 可以用整型二维数组表示 URL 和 keyword 的关系, 1 表示存在关系, 0 则无关系.

本文采用自底而上的批处理算法, 利用成员搜索引擎返回结果中的 URL 与标题/摘要信息分词来构造概念格. 每个搜索引擎 A 的概念格由二元组的概念节点构成, 其二元组为 $(\mathrm{URL}_A, \mathrm{keyword}_A) = (\{\mathrm{URL}_{A1}, \mathrm{URL}_{A2}, \ldots, \mathrm{URL}_{An}\}, \{\mathrm{keyword}_{A1}, \mathrm{keyword}_{A2}, \ldots, \mathrm{keyword}_{An}\})$. 对象集合 URL_A 与与属性集合 $\mathrm{keyword}_A$ 之间满足概念格相关定义中的约束. 包含 URL 数目相同的结点位于概念格的同一层. 层次越大, 结点中包含的 URL 越多.

计算两个搜索引擎属性关键词集合 $\mathrm{keyword}_A = \{\mathrm{keyword}_{A1}, \mathrm{keyword}_{A2}, \ldots, \mathrm{keyword}_{An}\}$ 和 $\mathrm{keyword}_B = \{\mathrm{keyword}_{B1}, \mathrm{keyword}_{B2}, \ldots, \mathrm{keyword}_{Bn}\}$ 的相似度利用笛卡尔积公式如式 (4):

$$\mathrm{weight}(\mathrm{keyword}_A, \mathrm{keyword}_B) = \frac{1}{n \times m} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathrm{distance}(\mathrm{keyword}_{Ai}, \mathrm{keyword}_{Bj}). \tag{4}$$

概念 $A(\mathrm{URL}_A, \mathrm{Keyword}_A)$ 和概念 $B(\mathrm{URL}_B, \mathrm{Keyword}_B)$ 的距离计算公式如式 (5):

$$\mathrm{nodeDistance}(A, B) = \frac{|\mathrm{URL}_A \cap \mathrm{URL}_B|}{\max(\mathrm{URL}_A, \mathrm{URL}_B)} \times w + \mathrm{weight}(\mathrm{keyword}_A, \mathrm{keyword}_B), \tag{5}$$

$\max(\mathrm{URL}_A, \mathrm{URL}_B)$ 表示集合 URL_A 与集合 URL_B 的最大的势力, w 是权重因子, $0 < w < 1$, 为偏向词语相似度的权重, 我们将 w 设为 0.8, 对于任意概念 A 和 B 均有 $\mathrm{nodeDistance}(A, B) = \mathrm{nodeDistance}(B, A)$. 当用户提交查询给不同的搜索 API, 获得 URL、标题、摘要等信息之后构造相应的概念格.

不同的概念格之间的层数可能不相等, 定义层数 L=min(L(A), L(B)), 最大节点 $\max(A)$, $\max(B)$, 最小节点为 $\min(A)$, $\min(B)$. $\max_i(A)$ 是在概念格 A 的第 i 层的概念结点中, keyeword 属性集合与查询词相似度最大的概念结点. 计算概念格 A 和概念格 B 的距离公式如式 (6):

$$\begin{aligned} \mathrm{conceptDis}(A, B) =& \mathrm{nodeDis}(\max(A), \max(B)) + \mathrm{nodeDis}(\min(A), \min(B)) \\ &+ \sum_{i=1}^{L-1} \mathrm{nodeDis}(\max_i(A), \max_i(B)), \end{aligned} \tag{6}$$

$\mathrm{conceptDis}(A, B)$ 可以理解为 B 对 A 的支持度, 同理可以得到其他概念格对概念格 A 的支持度 $\mathrm{conceptDis}(A, C)$, $\mathrm{conceptDis}(A, D)$ 等. 最后将所有概念格对概念格 A 的支持度求和得到概念格 A 最终支持度, 即搜索引擎 A 最终的权重如式: $\mathrm{Weight}(A) = \sum_{i=B}^{F} \mathrm{conceptDis}(A, B)$.

选取若干属于不同兴趣主题的关键词, 计算获取搜索引擎权重. 然后按照主题类别划分, 将同一类别中的所有权重求平均值即可得到搜索引擎对该兴趣主题的权重, 即擅长程度. 由此同理构造搜索引擎评价矩阵, 计算所有成员搜索引擎对不同兴趣主题的擅长程度.

5.2.2 基于日志分析的搜索引擎评价方法

搜索引擎日志记录了用户查询的信息, 如查询词、查询时间、第几次点击、该网页在返回中的排

1. After the user submits query string X, m concept lattices are generated and score A based on concept lattice for each member search engine is calculated.
2. After decomposing the query string into words, the category of X can be determined and the score B based on log analysis is calculated.
3. The score A and B, score B should be unified. So the score B is processed firstly. Score B should divide the sum of all score B and then the result multiply by the sum of all the score A. The final score of a member search engine can be calculated by the following formula:

$$\text{Weight} = B \times \left(\frac{\sum_{i=1}^{m} A_i}{\sum_{i=1}^{m} B_i} \right) + A.$$

4. Calculate scores for all member search engines for the query string X. In schedule process, the engine with higher score will be chose firstly.

图 4 基于日志与概念格的搜索引擎调度方法

Figure 4 Scheduling methods of search engine based on user log and concept lattice

名等, 这些信息反映了用户对该搜索引擎的满意程度. 利用这些信息设计出基于日志分析的方法, 来衡量用户对搜索引擎的偏好程度等方面的搜索能力的评价. 下面是基于日志分析的搜索引擎评价过程.

首先根据查询内容将日志中每条记录归类到不同的兴趣主题中, 利用本文提出的改进的基于词频的拉普拉斯平滑后的朴素贝叶斯算法对该条记录中用户点击页面进行文本分类, 从而确定查询内容、该条日志记录所归属的兴趣主题类别. 对日志中的每条记录按照公式: $W = 0.01 \times (100 - \text{rank})/\text{click}$ 进行计算, 得到针对该条记录, 搜索引擎对兴趣主题的权重得分.

click 表示用户第几次点击到该记录的 URL, rank 表示该记录中 URL 在搜索引擎返回检索结果中的排名. 考虑到用户点击的次数才最能说明用户想要的结果, 所以公式中除以 click, 则 click 越小 weight 越大. rank 排名越前, 则说明该结果越好, 用 100-rank 是考虑到用户几乎不去点击返回的结果中排名太靠后的 URL, 该处设置为若被点击结果排名超过 100, 则权重为 0.

若搜索引擎日志下共有 m 个查询内容属于某兴趣主题, 则该搜索引擎对该兴趣主题的权重得分为全部记录分值的平均值, 计算公式为 $\text{weight} = \frac{1}{m} \sum_{i=1}^{m} W_i$.

将搜索引擎所有日志记录分析计算, 可以得到该搜索引擎对所有兴趣主题的得分. 同理, 可以得到所有成员搜索引擎对所有兴趣主题类别的权重得分. 至此, 可以成功建立代表了用户使用偏好的搜索引擎评价矩阵.

5.2.3 基于概念格与日志分析结合的搜索引擎评价调度策略

结合前面提出的基于概念格与日志分析的评估方法, 综合权衡两个方法的权重, 本文提出了一种结合概念格和日志分析的搜索引擎评估调度算法. 对用户提交的搜索词返回的搜索引擎调度结果, 突出了搜索引擎兴趣擅长以及用户对搜索引擎的偏好程度的特点, 以此来衡量成员搜索引擎的搜索能力, 为制定调度策略提供依据, 从而可为实现个性化搜索功能提供支持. 结合概念格与日志分析的搜索引擎调度方法的具体步骤如图 4 所示.

6 实验研究及结果分析

本节对基于 Agent 的智能元搜索引擎个性化功能进行了实验研究与测试, 并对测试结果进行了分析. 本文以项目组开发的智能元搜索引擎 “智搜” 为测试平台, 从复杂查询语句识别功能测试、基于概念格与日志分析结合的搜索引擎评估调度、检索结果相关性三个方面进行了测试. 通过对测试结果的分析, 得出了本文提出的基于 Agent 的智能元搜索引擎个性化功能具有良好个性化搜索效果的结论.

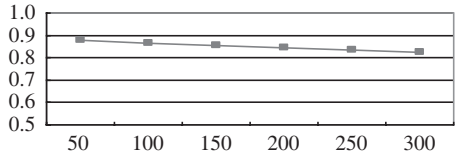


图 5 复杂查询语句识别正确率

Figure 5 The recognition accuracy of sophisticated retrieval condition

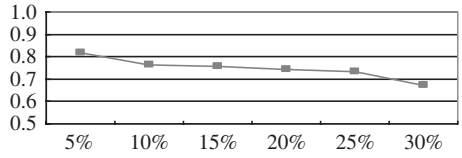


图 6 新复杂语义结构的学习情况

Figure 6 Learning situation of new SSS

项目组实现的元搜索引擎 “智搜”, 英文名为 “iSearch”, 包括网页搜索、社区搜索、学术搜索和电子商务搜索, 目前阶段只实现了网页搜索. 网页搜索的结果来自于百度、有道、搜搜、搜狗、雅虎、必应 6 个成员搜索引擎检索结果的合成, 由于谷歌中国搜索尚不稳定, 因此测试中并没有将其作为成员搜索引擎调用. 元搜索引擎系统 “智搜” 的运行环境如下: 系统部署在实验室的惠普服务器上, CPU 为 Inter(R) Core (TM) i3-2120 CPU@3.30 GHz, 内存是 4 G, 硬盘为 1 T. 操作系统是 Windows 7, Web 服务器容器为 Apache Tomcat 6.0, 数据库为 Mysql 5.0, Java 运行环境为 jdk 1.6.0_12. 客户端为普通 PC 机, Windows XP 操作系统, 主频 1. 6 GHz, 内存 1 G, 安装浏览器, 网络带宽为 10 Mbps. 后续测试中访问 “智搜” 系统的 PC 机安装 Chrome, Firefox 等主流浏览器, 网络带宽为 10 Mbps.

6.1 复杂查询语句识别功能效果测试

为了验证本文提出的基于动态学习机制的复杂语义识别方法, 我们首先检验了复杂查询语句提出的必要性, 即考察在信息检索时复杂查询语句出现的概率及其对检索结果好坏的影响情况. 具体实验过程见附录 E. 实验结果显示, 查询语句复杂度和复杂语义结构是影响搜索引擎检索效果好坏的两个重要因素. 因此具有其中一个或同时具有两个因素的复杂查询语句, 会对搜索引擎的检索效果造成较坏的影响. 我们将搜狗实验室提供的用户一天检索日志人工分为两部分, 一部分为简单查询语句, 另一部分为复杂查询语句. 从两部分中各自随机挑选查询语句, 分别合成 50 , 100, 150, 250, 300 个简单查询语句和复杂查询语句相混合的测试查询集合, 验证本文提出的识别算法的识别率. 实验结果如图 5 所示.

由图 5 可以看出, 在查询语句集合为 50 个的时候识别效果显示最好, 识别率接近 90%, 随着测试集合元素的增多识别率逐渐下降. 主要原因是测试集合中出现了一些没有被正确分词的专有名词和习语等词, 如 “猪元涨”、“米你屋”、“蒜你狠” 等, 需要人工对分词库进行补充. 但即便如此总体效果仍比较理想, 各组测试集合的识别率都在 80% 以上.

接下来验证对复杂语义结构 SSS 集合中不存在的新的复杂结构元素的学习效果, 随机选取 300 个检索条件, 在其中添加冷僻复杂语义结构如 “奥巴马竞选, 请问是哪一天”. 这种结构的出现频率很低, 但本文为测试需要均将其加入到测试集合中, 分别添加 5%, 10%, 15%, 20%, 25%, 30% 的新的复杂语义结构进行测试, 测试结果如图 6 所示.

图 6 显示在测试集合中不同比例数量新复杂语义结构的学习成功率, 新复杂语义结构数量在 5%~25% 时学习成功率可以达到 70% 以上, 在 30% 时学习成功率有所下降, 为 68.3%. 原因是本文提出的学习机制是以原有 SSS 集合中的元素为基础, 通过选取与集合元素相似度较大的新结构添加集合, 对于与集合中元素结构相差较大的新的复杂语义结构难以较好地识别, 但此种类型的语义结构在用户的检索条件中所占比例较低, 对本文提出的学习机制影响较小.

表 1 基于概念格的搜索引擎能力评价矩阵

Table 1 Evaluation matrix of ability of search engines based on concept lattice

	Education	Employment	Financial	Health	IT	Culture	Military	Sport	Tourism	Automobile
Baidu	1.676	1.413	1.533	1.353	1.652	1.486	1.295	1.578	1.329	1.628
Youdao	1.328	1.529	1.603	1.414	1.238	1.394	1.530	1.608	1.512	1.295
Soso	1.239	1.437	1.175	1.567	1.512	1.683	1.594	1.187	1.465	1.495
Sogou	1.321	1.638	1.236	1.346	1.454	1.534	1.641	1.561	1.682	1.349
Bing	1.695	1.353	1.416	1.287	1.247	1.435	1.192	1.358	1.305	1.344
Yahoo	1.345	1.136	1.390	1.348	1.601	1.346	1.645	1.615	1.163	1.689

表 2 基于日志分析的搜索引擎评价矩阵

Table 2 Evaluation matrix of search engines based on log analysis

	Education	Employment	Financial	Health	IT	Culture	Military	Sport	Tourism	Automobile
Baidu	0.885	0.921	0.853	0.815	0.912	0.8672	0.990	0.727	0.873	0.851
Youdao	0.978	0.975	0.960	0.974	0.981	0.973	0.970	0.960	0.975	0.782
Soso	0.985	0.900	0.981	0.780	0.985	0.788	0.985	0.818	0.759	0.926
Sogou	0.986	0.986	0.986	0.985	0.987	0.865	0.986	0.990	0.986	0.885
Bing	0.813	0.835	0.975	0.809	0.982	0.825	0.791	0.980	0.773	0.863
Yahoo	0.921	0.913	0.976	0.934	0.980	0.901	0.976	0.986	0.916	0.797

6.2 基于日志与概念格方法的成员搜索引擎的调度测试

基于日志与概念格方法的成员搜索引擎调度策略能对成员搜索引擎进行动态调度, 为检索用户提供个性化服务, 从而提升检索结果的质量, 下面我们的测试就是为了体现基于日志与概念格方法的优势. 本实验验证目标为使用百度、有道、搜狗、必应、雅虎、搜索多个搜索引擎同时调度情况下, 综合考虑了各种搜索引擎的专长, 以及词语相似度的比较, 针对不同搜索请求所做出的成员搜索引擎权重调整及返回结果的排序优先度. 本实验基于我们团队开发的智能元搜索引擎平台 “智搜”, 每次调度的结果是给出 3 个最优的成员搜索引擎. 由于首先要验证主题分类的准确性, 对文本分类方法进行了测试, 具体可见附录 F. 分类测试的实验结果显示不同主题类别分类准确率在 71.4%~92.6% 之间, 其中体育、军事、IT 等类别的成功率较高, 算法平均分类准确率达到了 82.06%, 对每篇文本文档的平均分类时间为 228.99 ms, 基本达到了实际应用的标准.

6.2.1 基于概念格的搜索引擎的评价方法测试

实验从文本分类训练集的每个分类中随机抽取 50 个具有代表性的词条, 向百度、有道、搜狗、搜搜、必应、雅虎等 6 个搜索引擎发起检索请求, 取前 20 条检索结果作为构造概念格的原始数据. 根据前面 5.2.1 小节中介绍的方法构造概念格并进行计算, 最终获得搜索引擎能力评价矩阵如表 1 所示, 数据保留小数点后 3 位.

6.2.2 基于日志分析的搜索引擎评价方法测试

“智搜” 后台保存了用户的搜索日志, 其中记录来自成员搜索引擎百度、有道、搜狗、搜搜、必应、雅虎, 从中随机选取 100 个用户 10 天的 6000 条搜索记录, 依据前面 5.2.2 小节中的方法进行计算, 最终得到基于日志分析的搜索引擎评价矩阵如表 2 所示, 数据保留小数点后 3 位.

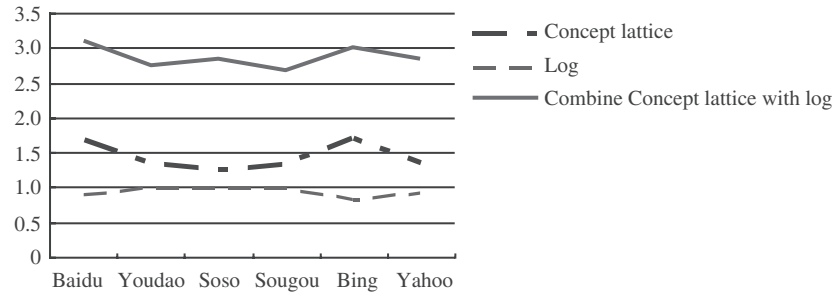


图 7　基于不同算法各搜索引擎得分对比

Figure 7　Scores comparison of different search engine

6.2.3　调度策略生成过程实例说明

在前面两节的实验中系统得到两个搜索引擎评价矩阵, 接下来用一个搜索实例说明调度策略的生成过程.

以用户搜索 “高考” 为例, 根据前面调度策略的设计, 系统先在用户兴趣模型中查找是否存在关键词 “高考”, 结果为不存在; 然后系统在文本分类训练库中查找关键词 “高考”, 发现高考存在于教育、就业、健康、IT、文化等多个类别中. 由于在教育类别中, “高考” 的词频最高, 因此搜索词 “高考” 属于主题类别 “教育”.

根据前面两节中得到的两个评价矩阵分别查找教育类下的各个搜索引擎的得分, 然后根据图 4 中的调度算法步骤计算得到最终得分. 因此可以得到图 7 中基于概念格、基于日志以及结合日志与概念格评估的三种算法各搜索引擎得分折线对比图. 元搜索最终调度得分最高的百度、必应和搜搜作为提供检索结果.

图 7 中通过对比, 得出针对查询词 “高考” 的基于日志评价得分相对平稳, 因为受日志数量的影响, 实验中的每个引擎的日志基本在 1000 条左右, 数量太少不足以得出具有代表性的评价, 而基于概念格的评价则差异比较大, 对搜索引擎给出了差异化的评分. 基于日志分析的搜索引擎的评分差别不大而结合日志分析与概念格的最终评分倾向于概念格的得分, 为了得出更有代表性的评价, 需要得到大量的日志数据进行测试, 这样结果才会更可靠.

本次实验中对 6 个搜索引擎一次检索结果构造概念格并计算的时间约为 7 min, 主要时间花费在查询词语相似度上. 由于构造搜索引擎评价矩阵的计算全部离线进行, 故并不影响元搜索引擎系统的正常使用. 当前搜索日志记录仍不够多, 数据不够具有代表性, 因此还需要取得大量的搜索日志进行实验.

6.3　检索结果相关性测试

本文设计的第三部分实验是对检索结果相关性的测试. 通常来说, 个性化的检索结果比非个性化的结果往往更加符合用户的查询意图, 因此检索相关性更高. 检索结果相关性测试的目的就在于考察本文提出的个性化功能是否具有学习能力, 学习效果如何以及在学习到了用户查询兴趣之后能否有效地将用户检索兴趣应用到查询结果的处理过程中, 并根据搜索能力对成员搜索进行合理调度, 为用户提供更加相关、全面的检索结果. 为了使测试结果更具有说服力, 需要选出有多重含义的代表性查询语句.

表 3 多义词语义类别及主流搜索引擎搜索结果比例

Table 3 Ratio of semantic category of polysemous words and serach results of mainstream search engines

Order	Polyseme	Category 1	Proportion (%)	Category 2	Proportion (%)
1	Lincoln	US president	21	Automobile	79
2	Amazon	IT company	78	Tropical forests	17
3	Xidian	Xidian university	69	XD group	20
4	Apple	IT company	77	Fruit	23
5	Kangxi	Emperor of Qing dynansty	40	TV program	60
6	Cloth-wrappers	Cloth bag	10	Crosstalk kokes	66
7	You are the one	Film	17	TV program	82
8	Jaguar	Automobile	89	Animal	11
9	Java	Programming language	91	Island of indonedia	9
10	GM	General Electric company	61	AMC	33

表 4 搜索结果排序数据

Table 4 Sorted data of the search results

User id	Interest category	Query word	Initial relevancy of results in first three pages (%)	Relevancy of results in first three pages after intelligent learing (%)	System response time (s)
1,2,4,7,11	US president	Lincoln	20	60	1.2
2,3	Tropical forests	Amazon	16	73	1.1
5,6,7	Xidian university	Xidian	70	90	1.2
6,10,13,14	IT company	Amazon	78	89	1.3
		Apple	77	86	1.1
8,9	Crosstalk kokes	Cloth-wrappers	68	88	1.2
7,12,15	TV program	Kangxi	71	93	1.2
		If you are the one	68	95	1.2
8,9	Animal	Jaguar	10	78	1.3
6,14	Island of indonesia	Java	9	79	1.4
8,10	AMC	GM	35	91	1.2
	Mean value		47.5	83.8	1.2

6.3.1 用户兴趣学习效果测试

在实验中, 本文首先选取了 10 条常见的多义词查询语句, 分析了它们的语义类别. 然后将所有查询提交至主流的搜索引擎如百度、谷歌等进行检索, 统计属于各个类别的检索结果的个数并计算其平均出现比例. 因研究表明用户大多只浏览前 30 个结果 [12], 因此本文只统计了所有查询语句在主流搜索引擎的检索结果前 3 页中出现的比例. 最终统计情况如表 3 所示.

由表 3 可以看出每条多义查询在主流搜索引擎中的检索结果情况, 检索结果页面中存在着语义类别不同的检索结果, 而且各自占有一定的比例数量. 我们选取对表 3 所述的 10 类多义查询有清晰兴趣类别的 20 名参与者进行实验, 要求他们在一周内多次在 “智搜” 中检索表 3 中的多义查询语句. 在对表 3 中查询进行检索并统计了检索结果相关程度、推荐查询相关程度及响应时间数据后, 得到表 4 的统计结果.

由表 4 可以看出, 初始状态下前三页查询结果的平均相关程度为 47.5%, 经过一周的智能化学习后 “智搜” 前三页查询结果的平均相关程度达到 83.8%. 对于印尼岛屿 “Java”、动物 “美洲豹” 等初始状态下前三页结果的相关程度极低的查询词, 在一周学习后, 相关度也达到 80% 左右, 体现了系统良好的学习效果. 虽然一开始在没有了解用户查询兴趣倾向的前提下无法提供个性化的检索结果, 但通过对用户浏览与点击行为的记录分析, 在一段时间之后便可初步判断出用户的查询兴趣. 进而利用该

表 5　中文搜索引擎的相关性

Table 5　Correlation of Chinese search engine

iSearch	Baidu	Google China	Yahoo China	Sohu	Youdao	YuanSearch	JoPee
0.6832	0.5998	0.5857	0.5612	0.4786	0.4322	0.6732	0.6632

表 6　英文搜索引擎的相关性

Table 6　Correlation of English search engine

iSearch	Google	Yahoo	InfoSeek	Bing	AltaVista	Metacrawler	Dogpile
0.7554	0.7333	0.7244	0.7105	0.6788	0.6324	0.7131	0.6879

兴趣进行检索结果的过滤, 为每个用户提供有针对性的检索结果, 使其更符合用户的检索意图, 实现个性化的检索.

同时, 系统的平均响应时间为 1.2 s, 该时间为服务器从接受搜索请求到向用户返回搜索结果间经历的时间间隔. 虽然由于元搜索引擎架构本身的原因导致其比传统搜索引擎的响应时间略长, 但仍在用户可接受的范围之内. 通过以上实验表明, 本系统具有有效的个性化功能和良好的性能.

6.3.2　检索效果对比测试

为了进一步考察本文所述的智能元搜索引擎个性化技术的优势, 本文选取了 14 个中、英文搜索引擎、元搜索引擎与 “智搜” 进行检索效果的对比测试. 其中中文搜索引擎包括百度、谷歌中国、雅虎中国、搜狐、有道, 中文元搜索引擎包括元搜、JoPee, 英文搜索引擎包括 Google, Yahoo, InfoSeek, bing, AltaVista, 英文元搜索引擎包括 metacrawler、dogpile, 在实际 Web 环境中进行了检索效果对比测试实验. 因为商业引擎是高度优化的, 因此任何 (甚至相对较小) 改善都是结果上的一种显著的提升.

评测采用搜索行业通用的 Cranfield 评估框架, 根据查询词背后的用户需求, 判断某条结果的质量, 用不同的分值表示结果的好坏程度. 中文搜索评测集除了表 3 中的多义查询语句外, 还包括了如 “恒大”、“雾霾” 等热搜词和 “北京保安总公司丰台区分公司”、“将 word 设置成绿色背景 win 7” 等长尾词, 同时参考了搜狗实验室提供的文本分类语料中十大分类下的代表性语句. 每个结果相关性的计算参考了文献 [13] 中提出的相关性评价方法如公式: $R = \frac{2r+u}{2t}$.

其中 r, u 和 t 分别为相关文档、不确定文档和检索到的文档数. 实验测试了各搜索引擎检索结果的平均相关性, 并与各成员搜索引擎进行比较, 其实验结果见表 5 所示.

因有多义词及长尾词的存在, 所以 R 值均低于 0.7, 但 “智搜” 相比其他搜索引擎和元搜索引擎有更高的 R 值, 说明其显示了最全和相关性更好的检索结果.

英文搜索语句选取文献 [14] 里采用的典型检索语句 20 条, 分别对各搜索引擎进行搜索测试, 记录检索结果的相关文档 r、不确定文档 u 和检索到的文档数 t. 然后采用上面的评价公式计算各英文搜索引擎的相关性, 如表 6 所示.

英文搜索引擎的 R 值较中文搜索引擎高, 平均值在 0.7 左右. 最高的为 Google, 达到 0.7333, 最低的为 AltaVista, 为 0.6324. iSearch 以 0.7554 的 R 值高居所测试搜索引擎中的第一位, 说明其对英文查询同样具有良好的搜索能力.

由表 6 可以直观地看出 iSearch 的平均相关性要好于各个成员搜索引擎及元搜索引擎, 而且比 AltaVista 的相关性要高近 20%. 通过检索结果相关性测试可充分说明本文提出的元搜索引擎个性化功能可有效地提高检索效果, 用户兴趣学习效果测试说明本系统具有良好的用户查询兴趣学习能力, 检索效果对比测试说明本系统的搜索效果相关性要好于传统的搜索引擎和元搜索引擎.

7 结束语

本文主要讨论了大量数据环境下基于 Agent 的智能元搜索引擎个性化机制, 目的在于有效提高用户在海量数据环境下信息检索效率, 实现智能化的元搜索引擎, 为用户提供智能化的搜索服务. 由于搜索引擎智能化的研究内容非常广泛, 大数据的规模效应也给数据分析带来了极大的挑战, 本文仅做了很有限的工作, 而且存在一些不完善的部分, 后续仍有许多重要的研究内容值得开展. 例如基于本体的大规模语义数据分析挖掘的研究, 检索结果数据深度分析, 更好的挖掘用户兴趣, 给用户做出更好的智能精准推荐; 对复杂查询语句识别机制的研究, 在基本能够准确识别出复杂查询语句后应该可以自动化地将其转换为简单查询语句, 提高搜索引擎的检索精度. 时间复杂度的设计是 web 信息检索的瓶颈, 还需要对算法进行改进, 提高效率. 对检索结果的合成处理的研究, 在能够合成传统的检索结果之后, 可进一步研究能否对如 “快递查询”、“火车票查询” 等 “事务类” 的查询进行有效合成. 作者后续将会对以上问题做进一步的研究与探讨.

参考文献

1 Vesset D, Woo B, Morris H D, et al. Worldwide big data technology and services 2012–2015 forecast. IDC Rep, 2012, 233485

2 Zeng C, Xing C X, Zhou L Z. A personalized search algorithm by using content-based filtering. J Softw, 2003, 14: 999–1004 [曾春, 邢春晓, 周立柱. 基于内容过滤的个性化搜索算法. 软件学报, 2003, 14: 999–1004]

3 Pretschner A, Gauch S. Ontology based personalized search. In: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, Chicago, 1999. 391–398

4 Metzler D, Croft W B. Analysis of statistical question classification for fact-based questions. Inform Retrieval, 2005, 8: 481–504

5 Luo Z S, Zheng B X. The research on algorithm and strategy of analysis automatically and distribution statistics for chinese sentence pattern. J Chinese Inform Process, 1993, 8: 1–19 [罗振声, 郑碧霞. 汉语句型自动分析和分布统计算法与策略的研究. 中文信息学报, 1993, 8: 1–19]

6 Xi J Q, Luo Q. Research on automatic identification for chinese prepositional phrase based on HMM. Comput Eng, 2007, 33: 172–173 [奚建清, 罗强. 基于 HMM 的汉语介词短语自动识别研究. 计算机工程, 2007, 33: 172–173]

7 Wu W S, Liu K L, Meng W Y, et al. A statistical method for estimating the usefulness of text databases. IEEE Trans Knowl Data Eng, 2002, 14: 1422–1437

8 Beg M M. A subjective measure of web search quality. Inform Sci, 2005, 169: 365–381

9 Wooldridge M, Jennings N R. Intelligent agents: theory and practice. The Knowl Eng Rev, 1995, 10: 115–152

10 Jennings N R. Agent-Oriented Software Engineering. Multiple Approaches to Intelligent Systems. Berlin: Springer, 1999. 4–10

11 Jeh G, Widom J. Scaling personalized web search. In: Proceedings of the International World Wide Web Conference, New York, 2002. 271–279

12 Jansen B J, Spink A, Bateman J, et al. Real life information retrieval: a study of user queries on the web. In: Proceedings of the ACM SIGIR Forum, New York, 1998. 5–17

13 Keyhanipour A H, Moshiri B, Piroozmand M, et al. WebFusion: fundamentals and principals of a novel meta search engine. In: Proceedings of IEEE International Joint Conference on Neural Networks, Vancouver, 2006. 4126–4131

14 Keyhanipour A H, Moshiri B, Kazemian M, et al. Aggregation of web search engines based on users' preferences in WebFusion. Knowl-Based Syst, 2007, 20: 321–328

补充材料 附录 A ∼ F. 本文的补充材料见网络版 info.scichina.com. 补充材料为作者提供的原始数据, 作者对其学术质量和内容负责.

Personalization mechanism in an intelligent agent-based meta-search engine

LI QingShan[1,2,3*], WANG Jun[1,3], CHU Hua[1,2] & JI TaoRan[1,3]

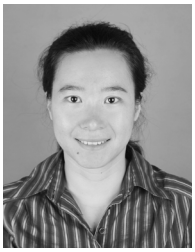
1 *Software Engineering Institute, Xidian University, Xi'an* 710071*, China*;
2 *School of Software, Xidian University, Xi'an* 710071*, China*;
3 *School of Computer Science and Technology, Xidian University, Xi'an* 710071*, China*
*E-mail: qshli@mail.xidian.edu.cn

Abstract Information overload is a common issue in "big data" environments, because of which such environments require precise and intelligent retrieval tools. In this paper, we propose the design and implementation of a personalization function for an intelligent, agent-based meta-search engine that improves information retrieval efficiency in "big data" environments. We focus on individualized methods, along with the relevant theoretical and technological requirements for the implementation of a personalization function in an intelligent, agent-based meta-search engine, provide query analyses, and describe the interest mining and scheduling processes of the search engine. We then detail the recognition mechanism of complex inquires based on a dynamic learning process, the interest mining mechanism based on the general view model of user interest, which involves dynamic updates, and the strategy for search engine scheduling based on a concept lattice and user logs. Experiments showed that the personalization function for intelligent, agent-based meta search engines proposed here accurately identified complex queries, effectively learned users' search interests, improved the relevance of the search results, and intelligently scheduled member search engine. It also helped improve the efficiency of information retrieval by users, thus enhancing their searching experiences.

Keywords information overload, meta-search engine, personalization, agent, relevance of search results

LI QingShan was born in 1973. He received his Ph.D. in computer application technology from the Institute of Software Engineering of Xidian University, Xi'an, China, in 2003. He is currently a professor at Xidian University. His research interests include agent technology, software evolution, software architecture, program analysis, and meta-search technology. Prof. Li is an academic leader in software engineering at the provincial and ministerial levels, and enjoys the allowance of "San Qin Talent" in Shaanxi province. He is a member of the IEEE, ACM, and CCF.

WANG Jun was born in 1989. She is currently pursuing her Master's degree in computer software and theory from the Institute of Computer Science and Technology in Xidian University, Xi'an, China. Her research interests include information retrieval, agent technology, and software architecture.



CHU Hua was born in 1977. She received her Ph.D. from the School of Computer Science and Technology in Xidian University, Xi'an, China, in 2007. She is an associate professor at the Institute of Software Engineering at Xidian University. Her research interests include clinical decision support systems, software architecture, reverse engineering, and program analysis.

JI TaoRan was born in 1989. He is currently pursuing his Master's degree in computer software and theory at the Institute of Computer Science and Technology of Xidian University, Xi'an, China. His research interests include information retrieval and software architecture.