元搜索中成员搜索引擎的选择问题研究

刘登洪 徐 贤
(华东理工大学计算机科学与工程系 上海 200237)

摘 要 随着网络的普及,网上检索成为了人们获取信息的主要方式。目前的搜索引擎相对独立,覆盖范围比较有限。相比之下,元搜索能够更好地满足用户的检索需求。当用户在元搜索提供的统一界面中输入一个查询时,元搜索会将处理后的用户请求发送给相关的成员搜索引擎。但是一个重要的问题是如何识别出潜在的搜索引擎以便更好地处理用户的请求。鉴于此提出了一种基于遗传算法的选择机制,该方法将各个成员搜索引擎的权重考虑在内。实验结果表明,该方法确实能够提高引擎选择中的效率和精度。

关键词 元搜索,查询,引擎选择
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.10.042

Research on Member Search Engine Selection in Meta Search

LIU Deng-hong XU Xian
(Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

Abstract With the popularity of network, searching online becomes the main way to get information. Compared to independent search engine usually with limited coverage, meta search engine can meet the needs of information retrieval in a better way. When a query is input in the unified interface provided by meta search, it first processes the query and then sends it to appropriate member search engines. An important problem is how to find the underlying search engines which can optimally reply to the user query. In this paper, we proposed a mechanism based on genetic algorithm, which also takes the weight of each member search engine into account. The experimental results show that our method can indeed improve efficiency and accuracy on engine selection.

Keywords Meta search, Query, Engine selection

1 引言

元搜索引擎是一种检索工具,它能够同时将用户的查询请求发送给多个搜索引擎进行处理。元搜索具有覆盖范围广、易扩展、用户体验佳等特点,是网络信息检索领域的一个重要应用[1]。元搜索并不拥有自己的索引库,它建立在多个搜索引擎之上,这使得元搜索引擎能够获取到更多有价值的信息和更多综合、精确的结果[2]。

1.1 元搜索引擎的架构

元搜索引擎的架构如图1所示。一个元搜索引擎主要由以下5个部分组成:统一用户界面 User Interface、搜索引擎选择器 Search Engine Selector、文档选择器 Document Selector、查询分发器 Query Dispatcher 和结果合并器 Result Merger。元搜索虽然没有自己独立的索引库,但是保存有一些成员搜索引擎的信息。当用户在 User Interface 中输入查询请求后,Search Engine Selector 会根据用户的请求推荐一组相关的引擎列表。在这些搜索引擎列表中的成员最有可能包含用户的查询内容。Document Selector 决定了哪些文档需要从列表中的成员搜索引擎返回。Query Dispatcher 负责将查询分发到各个选定的成员搜索引擎。最后,各个选定的引擎返回的结果在 Result Merger 组件进行排序。排序后的结果为一个单一的列表并作为最终查询结果被推送到用户界面。

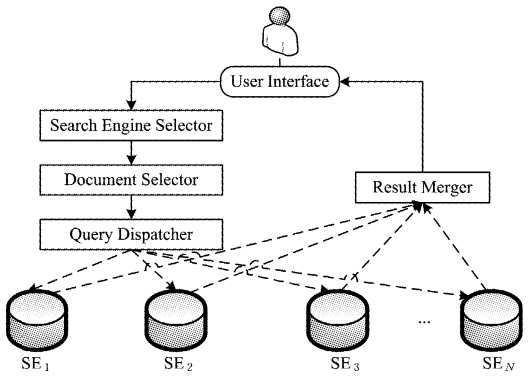

图1 元搜索引擎架构示意图

1.2 相关工作

Search Engine Selector 的主要功能是识别出与查询最相

到稿日期:2016-09-23 返修日期:2017-01-05
刘登洪(1991—),男,硕士生,主要研究方向为数据挖掘与知识发现等,E-mail:liudenghong@126.com;徐 贤(1979—),男,副教授,主要研究方向为并发理论及其应用等,E-mail:xuxian@ecust.edu.cn。

关的搜索引擎，但是如何科学合理地进行选择一直是一个挑战性的问题[3]。文献[4]提出的 CORI 方法中每个数据源使用一个“超级文档”表示。若一个词在数据库中出现在 k 个文档中，那么这个词在“超级文档”中的频率就标记为 k，因此一个词在数据库中出现的频率即为“超级文档”中的词频。然后利用“超级文档”中的词频和文档频率计算词的权重来进行查询。CORI 方法虽然稳定却并没有考虑数据源大小导致的“超级文档”差异。文献[5]提出一种 ReDDE 数据源选择方法，ReDDE 方法综合考虑了各个数据源的大小差异，同时提出一种抽样-再抽样的数据库大小评估方法，通过很少的信息交互就能做出较为准确的评估。文献[6]提出一种 Qsim 方法，强调利用以往查询的有用信息为具体的用户查询评估各个数据源的可靠性。遗传算法在文献检索中应用较早，但是以往多被应用于单一数据源中[7]。文献[8]将遗传算法用于多数据源的选择，同时运用一种新的突变方式，实验结果表明这种进化方法比 Qsim 更加高效；但是在计算搜索引擎和查询相关度时，该方法需要将查询发送至每一个成员搜索引擎，这无疑增大了选择的代价。

1.3　本文工作

通过对原有 Qsim 方法进行改进，提出一种 IQsim 方法。具体工作如下：

1）IQSim 方法结合了遗传算法 GA（Genetic Algorithm）和 Qsim 方法，能够显著提高各个成员搜索引擎的适应值 *Fitness*。

2）在对文档的合并选择中使用 Weighted Round Robin 方法，在避免重复计算相同文档的同时兼顾了各个搜索引擎的公平性。

3）在遗传算法中使用一种新的交叉方式。

2　Qsim 方法的改进设计与实现

研究 Qsim 的执行过程时，主要聚焦在用户查询与各个过往查询的相似性即 $sim(p_i|q)$ 的计算机制上。

2.1　Qsim 的原理

在现实的全文本检索系统中，很多相似的以往查询能够为我们提供很有价值的信息，帮助我们对数据源进行正确的选择。文献[6]的 Qsim 方法正是基于此思想，下面给出 Qsim 的计算步骤。

输入：H{S,P,q}，其中 S 为成员搜索引擎的集合，P 为选取的以往相似查询集合，q 为本次用户查询

输出：H′{S}，S 为适应值 Fitness 从大到小的成员引擎集合，此处 Fitness 指用户查询 q 和成员搜索引擎的相关度 $rel(s_j|q)$，Fitness 值越高，说明该搜索引擎越有可能包含用户的查询内容

步骤 1　计算各个搜索引擎和各个以往查询的相关性：

$$rel(s_j|p_i)=\sum_{T}\frac{rel(s_j|doc_t)}{T}$$

其中，T 是设定值，为每个以往查询的返回文档结果总数。doc_t 为返回的第 t 个文档，$H(p_i)$ 是第 i（$i\leqslant T$）个以往查询 p_i 的返回结果集合。如果 $doc_t\in H(p_i)$ 并且 $doc_t\in s_j$，则 $rel(s_j|doc_t)$ 为 1；否则为 0。

步骤 2　计算各个以往查询和用户查询的相似性 $sim(p_i|q)$。

步骤 3　计算各个成员引擎和用户查询的相关性：

$$rel(s_j|q)=\sum_{i}rel(s_j|p_i)*sim(p_i|q)$$

目前对 $sim(p_i|q)$ 的计算有很多方法，如文献[9]提出的余弦计算公式，其基于 p_i 和 q 的词频相似信息；文献[10]提出的基于二者查询结果重叠度的方法。但是文献[11]提出在合并的 T 个列表中可能存在相同的文档被重复用于计算 $sim(p_i|q)$，因此引入一种 Round Robin 的轮询计算，避免了相同的文档出现在同一次的合并之中。Round Robin 遇到权值相同的文档时总是选择第一次出现的文档，这种看似公平的做法实际上对来自其他引擎的文档并不公平。基于此，本文引入了加权的 Weighted Round Robin[12] 方法。

2.2　IQsim 方法的实现

本文提出的 IQsim 算法基于 Qsim。本节分两部分对该算法进行介绍：第一部分是算法的步骤；第二部分是算法在每一部分中所涉及的详细操作。

2.2.1　IQsim 方法的步骤

IQsim 方法的步骤如下。

输入：H{S,P,q}

输出：H′{S}

步骤 1　在搜索空间生成一组引擎向量。

步骤 2　计算每一个引擎向量的适应度 Fitness 值。

步骤 3　引入 GA 操作。由于每一个查询都有来自不同引擎的结果，在合并结果时我们使用 Weighted Round Robin 算法。对于权值的分配，可以通过本地分析用户的点击和浏览行为获取用户偏好从而实现个性化分配。本实验为简化并突出算法核心，采用了随机分配方法。

步骤 3.1　对每个引擎向量执行交叉操作。

步骤 3.2　如果引擎的向量 Fitness 值没有增加，则执行突变操作。

步骤 3.3　选择操作。只有 Fitness 值突变后增加的群体才有可能存活下来。

步骤 4　如果 Fitness 值可以继续增加则重复步骤 2 和步骤 3；否则按照各个引擎向量 Fitness 值的大小输出结果。

2.2.2　IQsim 方法中每一部分的具体实现

（1）初始化引擎向量

在 Qsim 方法[6] 中首先需要计算 $rel(s_j|p_i)$。由于每个 p_i 返回的都是一个合并列表，因此列表中的每一个文档都有可能来自不同的引擎。初始化一个 $M*N$ 的矩阵，其中 M 表示搜索引擎个数，N 表示每个引擎针对每个查询所返回的结果数。现实中对于同一个查询返回的结果数可能并不一样。P 指若干个以往查询集合，也需要进行初始生成。实验设置 P 为 $20*3$ 的矩阵，即代表 20 个以往查询，每个查询维度为 3。

（2）计算 *Fitness*

Fitness 的计算公式为：

$$rel(s_j|q)=\sum_{i}rel(s_j|p_i)*sim(p_i|q)$$

从上式可以看出，如果要计算出 *Fitness*，则首先需计算 $rel(s_j|p_i)$ 和 $sim(p_i|q)$，在此采用一种基于结果的半监督式计算方法[10]，详细说明如下。

1）$rel(s_j|p_i)$ 的计算

针对每一个以往查询，M 个搜索引擎返回了 $M*N$ 个结果。最终这些结果将会合并成一个单一的列表。为了避免重复结果出现在合并列表中，使用 Weighted Round Robin 算法进行结果的合并。具体合并规则是依次按列合并结果。当遇到相同的文档时，根据权值的大小，选择权值大的搜索引擎文档加入列表。

2)$sim(p_i|q)$的计算

将随机产生的用户查询在初始化向量矩阵中进行检索。由于初始化向量矩阵都是与用户查询集合 P 相关的返回结果,因此使用 q 查询后,得到的 T 个排序后的结果必定与 P 中的结果存在一定相似性。二者的相似性计算公式为:

$$sim(p_i|q)=\frac{1}{|R_{p_i}|}\sum_{doc\in R_{p_i}\cap R_q} score(doc,R_{p_i},R_q)$$

其中,$score(doc,R_{p_i},R_q)$是共同文档计分函数。

$$score(doc,R_{p_i},R_q)=1-\left|\frac{docRankInR_{p_i}}{|R_{p_i}|}-\frac{docRankInR_q}{|R_q|}\right|$$

其中,$|R_{p_i}|$是第 i 个以往查询返回的结果列表,$|R_q|$是用户查询 q 返回的结果列表。$docRankInR_{p_i}$ 和 $docRankInR_q$ 分别表示同一文档在$|R_{p_i}|$和$|R_q|$中的排列序号。

为便于计算,需要对 $sim(p_i|q)$进行标准化操作:

$\max sim=\max(sim(p_i|q))$

$cutsim=0.8*\max sim$

若 $sim(p_i|q)$小于 $cutsim$,则标准化后的 $Normsim$ 取值为 0,否则 $Normsim$ 取值为$\frac{sim(p_i|q)-cutsim}{\max sim-cutsim}$。

3)引入 GA

①交叉操作

将每一个引擎视为一条染色体,返回的每一个文档视为每一条染色体上的基因。实验中使用多项式交叉[13]:

$$X_i^{(1,t+1)}=0.5[(1+\beta)X_i^{(1,t)}+(1-\beta)X_i^{(2,t)}]$$

$$X_i^{(2,t+1)}=0.5[(1-\beta)X_i^{(1,t)}+(1+\beta)X_i^{(2,t)}]$$

其中,$X_i^{(1,t)}$ 和 $X_i^{(2,t)}$ 为父代个体;$X_i^{(1,t+1)}$ 和 $X_i^{(2,t+1)}$ 为交叉后的新生个体。β取值为:

$$\beta=\begin{cases}(2u)^{\frac{1}{n+1}}, & u\leqslant 0.5\\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{n+1}}, & u>0.5\end{cases}$$

其中,u 的随机取值区间为[0,1]。n 的取值反映了子代和父代关系的远近。n 的值按照文献[14]指定为 20。多次实验也证明当 $n=20$ 时效果最好。

这种多项式交叉产生的两个子代个体关于父代对称,保证了子代不会偏向于任一个父代,同时子代也继承了父代之间的距离特性。如果父代之间相差较远,则新生子代之间的距离不可能较近。

②突变操作

突变是指染色体中的一个基因值随机被新值替代,染色体中父代个体的突变发生概率通过突变参数设定。在实验中对于突变的发生,选择的是多项式突变[13]:

$$m=\begin{cases}P+\delta_L(P-X^L), & u\leqslant 0.5\\ P+\delta_R(X^R-P), & u>0.5\end{cases}$$

其中,$\delta_L=(2u)^{1/(1+n)}-1$,$\delta_R=1-(2(1-u)^{1/(1+n)})$。这里 m 即为突变后的个体;X^R 和 X^L 分别为文档的上限和下限;P 为父染色体;n 和 u 与交叉操作中的 n 和 u 为相同变量,取值相同。

这种多项式突变是在父代的基础上进行的。文档上限 X^R 和下限 X^L 的存在能够保证突变后的个体始终在合理的范围内。

③选择操作

选择操作基于最适应原则。将每次循环迭代后的 $Fitness$ 值与迭代前的进行对比,较大的值将留下来作为新的群体。

3 实验结果和分析

为了验证算法的有效性,在 Matlab2012a 上进行实验。由于文档检索采用的是向量空间模型,计算所需要的数据都是词频等数字信息,因此本实验参照文献[8],以模拟数据代替真实数据。初始化矩阵 $M*N$ 为 50*120,即 50 个搜索引擎,每个搜索引擎返回 120 个文档。同样,为了识别出与查询最相关的搜索引擎,设置循环 1000 次。

为了显示 IQsim 方法的有效性,设计了对比实验,将 IQsim 分别与文献[6]、文献[8]、文献[11]中方法进行比较,对比结果如图 2 所示。具有较高 $Fitness$ 值的搜索引擎会被推荐用于用户查询。$Fitness$ 值反映的是用户查询与各个搜索引擎之间的相关度的大小。实验结果表明,IQsim 方法不仅能够返回成员引擎较高的 $Fitness$ 值,而且在计算上比其他方法更加高效。

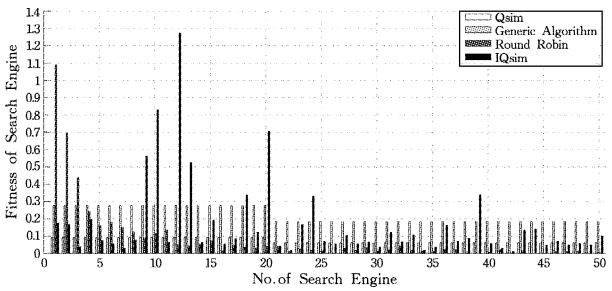

图 2 4 种方法的返回结果示意图

实验结果的具体分析如下:

1)从整体结果上来看,IQsim 能够处理搜索引擎数量较多的情况。图 2 中即使是第 39 个搜索引擎,IQsim 依然返回了较高的适应 $Fitness$ 值,这主要是因为 IQsim 中权值的选取是随机的。可以看出,如果权值从大到小依次递减,那么 Round Robin 所推荐的结果将会与 IQsim 相同。文献[8]的 Genetic Algorithm 由于没有设置权值,因此各个引擎的 $Fitness$ 值非常接近,很难识别出哪些引擎与查询请求最相关。而文献[11]的 Qsim 方法既没有设置权值,也没有额外的选择机制。因此 4 种方法中 Qsim 方法返回的 $Fitness$ 值最小。

2)从各种方法的推荐结果上看,对 4 种方法返回的 $Fitness$ 值从大到小排列比较,IQsim 返回的 $Fitness$ 值普遍较高,这主要是因为遗传算法在每次迭代中的选择作用,较高的 $Fitness$ 值总会被保留下来。

3)从效率上看,虽然 4 种方法均采用了遗传算法的选择机制,但是 IQsim 充分利用了以往查询及其结果。在查询和各引擎相关性的计算上,IQsim 直接基于以往的查询结果。而文献[8]的遗传算法需要将查询发送到所有成员引擎,无疑增大了额外开销。

结束语 本文主要研究了元搜索中成员搜索引擎的选择问题,提出了一种改进的 IQsim 方法。该方法引入了遗传算法的思想,利用以往相似查询等信息来计算各个搜索引擎和用户查询的相关性,并根据相关性的大小对各个搜索引擎进行排序;最后将用户查询发送给相关性高的搜索引擎进行文档检索。

(下转第 258 页)

[5] JINDAL N,LIU B. Opinion spam and analysis[C]// International Conference on Web Search and Data Mining. ACM,2008:219-230.

[6] OTT M,CHOI Y J,CARDIE C,et al. Finding Deceptive Opinion Spam by Any Stretch of the Imagination[C]// Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:Human Language Technologies. Stroudsburg,PA,USA: Association for Computational Linguistics,2011:309-319.

[7] LIU B. Sentiment Analysis and Opinon Mining[M]. Chicago: Morgan & Clayppol,2012:113-115.

[8] MA Y,LI F. Detecting review spam:Challenges and opportunities[C]// 2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom). IEEE,2012:651-654.

[9] DIAO Y F,LIN H F. LDA-based Opionion Spam Discovering [J]. Journal of Chinese Information Processing,2011,25(1):41-47. (in Chinese)
刁宇峰,林鸿飞. 基于 LDA 模型的博客垃圾评论发现[J]. 中文信息学报,2011,25(1):41-47.

[10] LAI C L,XU K Q,LAU R Y K,et al. Toward a language modeling approach for consumer review spam detection[C]// 2010 IEEE 7th International Conference on e-Business Engineering (ICEBE). IEEE,2010:1-8.

[11] JIN J,JI P. Co-training Algorithm for Quality Analysis of Online Customer Reviews[J]. Journal of Shanghai University(Natural Science Edition),2014,20(3):289-295. (in Chinese)
靳健,季平. 用于在线产品评论质量分析的 Co-trainning 算法[J]. 上海大学学报(自然科学版),2014,20(3):289-295.

[12] 中科院分词系统[DB/OL]. http://ictclas.org.

[13] XU L H,LIN H F,PAN Y,et al. The structure of the emotional vocabulary ontology[J]. Journal of Emotion,2008,27(2):180-185. (in Chinese)
徐琳宏,林鸿飞,潘宇,等. 情感词汇本体的构造[J]. 情感学报,2008,27(2):180-185.

[14] BLEI D M,NG A Y,JORDAN M I. Latent dirichlet allocation [J]. Journal of Machine Learning Research,2003(3):993-1022.

[15] QIU Y F,WANG J K,SHAO L S. Research on Product Review Spammer Detection Based on Users' Behavior[J]. Computer Engineering,2012,38(11):254-257. (in Chinese)
邱云飞,王建坤,邵良杉. 基于用户行为的产品垃圾评论者检测研究[J]. 计算机工程,2012,38(11):254-257.

(上接第 236 页)

当搜索引擎的数目较多时,如何高效地选出最适合的搜索引擎依然是一个挑战性的问题。目前人工神经网络已经被应用于引擎选择中,并已被证明具有有效性[15]。更进一步,我们将考虑不同的启发式方法和神经网络在引擎选择中的结合使用。

参 考 文 献

[1] MENG W Y,YU C,LIU K L. Building efficient and effective meta-search engines[J]. ACM Computing Surveys (CSUR),2002,34(1):48-89.

[2] XUE Y,SHEN X P,CHEN J B. Research on an Algorithm of Metasearch Engine Based on Personalized Demand of Users [C]// 2010 International Forum on Information Technology and Appli- cations(IFITA). IEEE,Kunming,China,2010:240-243.

[3] SUN Y C,LI Q S. The research situation and prospect analysis of meta-search engines[C]// 2012 2nd International Conference on Uncertainty Reasoning and Knowledge Engineering (URKE). IEEE,Bali,Indonesia,2012:224-229.

[4] CALLAN J P,LU Z H,CROFT W B. Searching distributed collections with inference networks[C]// Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, Massachusetts, USA,1995:21-28.

[5] SI L,CALLAN J. Relevant document distribution estimation method for resource selection[C]// Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, Toronto, Canada, 2003:298-305.

[6] CETINTAS S,SI L,HAO Y. Learning from past queries for resource selection[C]// Proceedings of the 18th ACM Conference on Information and Knowledge Management. ACM, Hong Kong, China,2009:1867-1870.

[7] RAVI S,GANESAN N,RAJU V. Search Engines Using Evolutionary Algorithms[J]. International Journal of Communication Network Security,2012,4(1):39-44.

[8] KUMAR R,SINGH S K,KUMAR V. A heuristic approach for search engine selection in meta- search engine[C]// 2015 International Conference on Computing,Communication & Automation(ICCCA). IEEE,Uttar Pradesh,India,2015:865-869.

[9] SHEO D,KULDEEP S R. Search Engine Selection Approach in Metasearch Using Past Queries[J]. Oriental Journal of Computer Science & Technology,2014,3(23):177-183.

[10] SI L,CALLAN J. A semisupervised learning method to merge search engine results[J]. ACM Transactions on Information Systems (TOIS),2003,21(4):457-491.

[11] KUMAR R,GIRI A K. Learning based approach for search engine selection in meta-search engine[J]. International Journal of Engineering and Management Research,2013,10(3):82-88.

[12] XU Z Y,WANG X X. A predictive modified round robin scheduling algorithm for web server clusters[C]// 2015 34th Chinese Control Conference(CCC). IEEE,Hangzhou,China,2015:5804-5808.

[13] DEB K,GOYAL M. A combined genetic adaptive search (GeneAS) for engineering design[J]. Computer Science and Informatics,1996,26(16):30-45.

[14] GORDON M. Probabilistic and genetic algorithms in document retrieval[J]. Communications of the ACM,1988,31(10):1208-1218.

[15] GOEL P,JAIN T,BHATIA M P S. Learning from training query in Meta search using Artificial neural network[C]// 2015 Annual IEEE India Conference (INDICON). IEEE, Delhi, India, 2015:1-6.