基于主题 Hub 值的元搜索

蒋宗礼，李宪雷，徐学可
（北京工业大学 计算机学院，北京 100124）

摘 要：为了提高元搜索引擎排序结果的质量，提出了成员引擎特征的主题 Hub 值表示和基于主题 Hub 值的结果排序算法．特征学习算法利用一组主题关联词对成员引擎的特征进行学习，并表示为主题 Hub 值的形式．排序算法根据主题 Hub 值计算结果的全局相关度对结果进行排序．实验结果表明，该模型取得了更好的排序质量．

关键词：搜索引擎；元搜索；排序；主题 Hub 值
中图分类号：TP391.1 **文献标识码**：A **文章编号**：0254－0037(2009)03－0397－06

由于互联网上的信息量巨大，没有一个搜索引擎可以覆盖所有的网页．为了获得所需的信息，人们有时不得不使用多个搜索引擎．元搜索引擎集中了多个搜索引擎的优势．传统的元搜索结果排序算法有引擎优先、摘要排序、位置排序．针对以上算法存在结果相关度差、排序序列不准确等不足，作者提出了基于主题 Hub 值的元搜索模型，根据成员引擎主题 Hub 值进行结果排序．

1 总体框架

基于主题 Hub 值的元搜索分为 3 个部分：成员引擎特征学习、结果排序和派遣收集，如图 1 所示．学习部分实现成员引擎主题 Hub 值的学习，包括主题 Hub 值学习器和主题 Hub 知识库．结果排序利用主题 Hub 知识库中的主题 Hub 值向量作为成员引擎的特征表示对结果进行排序．派遣收集部分负责把用户查询表示为成员引擎能接受的形式，最后收集各成员引擎的返回结果并转化为统一的结果格式．

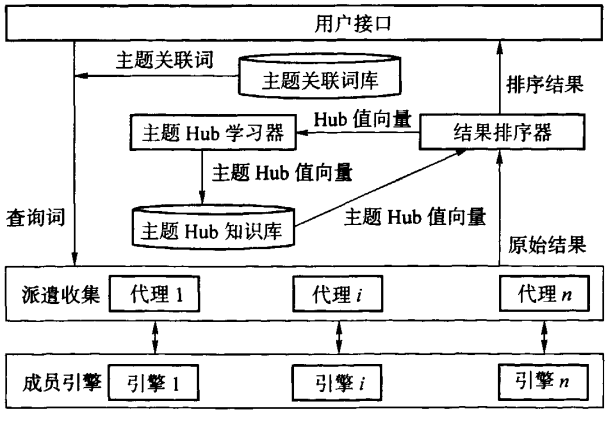

图 1 主题 Hub 元搜索总体框架

Fig.1 Architecture of topic hub based meta search engine

系统工作分为 2 个阶段：主题 Hub 值学习阶段和结果排序阶段．在学习阶段，从主题关联词库中提取

收稿日期：2007-10-25．
作者简介：蒋宗礼(1956－)，男，河南南阳人，教授．

某类主题的一组关联词,将这组关联词依次提交给元搜索,结果排序器根据返回结果计算成员引擎的一个Hub值向量,然后提交给主题Hub值学习器,最终学习到的主题Hub向量将添加到主题Hub知识库中.在结果排序阶段,结果排序器接收成员引擎返回的结果,然后从主题Hub知识库中获取相应的主题Hub向量,利用主题Hub向量计算结果的全局相关度,最后根据全局相关度对结果进行排序并返回给用户.

2 主题Hub值和结果排序

2.1 网页链接分析与排序

PageRank算法[1]利用网页之间链接关系计算网页的重要度.网页 i 的重要度

$$P(i)=d\times D(i)+(1-d)\times \sum [P(j)/N(j)] \tag{1}$$

式 $\sum$ 作用于所有链接指向网页 i 的网页 j,$N(j)$ 为网页 j 链接指向网页的总数.浏览者以概率 d 不再跟随链接浏览而重新从 $D(i)$ 分布中挑选一个新的网页重新开始浏览,以 $1-d$ 的概率从当前网页重新跟随一个链接浏览.谷歌采用这一技术成功提升了网页搜索的质量.

HITS算法[2]区分了Hub和Authority网页.Authority网页是主题相关的网页,而Hub网页包含了很多指向主题相关网页的链接,因此,Authority值表示网页的主题相关性,Hub值表示网页指向主题相关性网页的关联性.对于一个网页 i,x_i 表示它的Authority值,y_i 表示它的Hub值,即

$$x_i=\sum y_j \tag{2}$$

$$y_i=\sum x_j \tag{3}$$

经过式(2)、(3)迭代计算出每个网页的Hub值和Authority值,按照这2个不同的权值,分别取 k 个返回给用户.根据IBM研究院的Clever系统的测试数据,对于返回给用户的前10个检索结果,Clever系统在50%的情况下取得了高于搜索引擎雅虎和AltaVista的用户评价.

2.2 成员引擎Hub值和结果Authority值

成员引擎返回的结果序列一般是按照结果和查询词的相关度排序的[3],由于不同成员引擎计算相关度的方法不同,因此,不同成员引擎返回的结果之间的相关度是无法直接比较的,但它们都反映了成员引擎对结果的推荐:位置靠前的结果一定比靠后的结果好.给定成员引擎的一个结果序列 D 和一个结果 R,成员引擎对该结果的推荐度是关于 R 在 D 中位置 i 的单调递减函数

$$f(D,R)=\begin{cases}1/\mathrm{e}^i, & R\in D\\ 0, & R\notin D\end{cases} \tag{4}$$

由于不同引擎检索的性能不同,因此仅知道成员引擎对结果的推荐度,还是不能比较不同引擎返回的结果[4-5].例如,一个好的成员引擎靠后的结果不一定比一个不好的成员引擎靠前的结果差.因此要想比较不同结果之间的主题相关度,还必须考虑成员引擎自身的性能.

假设元搜索有 m 个成员引擎,它们返回的结果集合依次为 $D_1,D_2,\cdots,D_m$,并起来得到原始结果集 $\bigcup_{i=1}^{m}D_i=\{R_1,R_2,\cdots,R_n\}$,其中 R_i 表示元搜索的结果序列中的第 i 个结果.x_i 表示元搜索对第 i 个结果网页的全局推荐度,y_j 表示元搜索对第 j 个成员引擎的性能评价

$$x_i=\sum_{j=1}^{m}[y_j\times f(D_j,R_i)] \tag{5}$$

$$y_j=\sum_{i=1}^{n}[x_i\times f(D_j,R_i)] \tag{6}$$

如果把成员引擎返回一个结果的关系理解为成员引擎有一个指向该结果的链接,结果对应成员引擎的关系理解为结果指向成员引擎的一个链接,那么性能好的成员引擎指向的若干结果的相关度也较好,相关度好的结果也应该指向若干好的成员引擎.这样一种相互指向、相互评价的对偶关系[6]与超文本归纳主题搜索(hypertext induced topic search,HITS)算法中Hub页面和Authority页面的关系类似,故把上述

计算得到的结果的全局推荐度和成员引擎的性能评价分别称为结果的 Authority 值和成员引擎的 Hub 值. 把元搜索中成员引擎的 Hub 值和结果的 Authority 值分别表示为

$$\boldsymbol{Y}=(y_1,y_2,\cdots,y_m)^{\mathrm{T}},\boldsymbol{X}=(x_1,x_2,\cdots,x_n)$$

推荐度表示为推荐度矩阵

$$\boldsymbol{P}=\begin{pmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{pmatrix}$$

其中 $p_{i,j}$是第 i 个成员引擎对第 j 个结果的推荐度.

由此得到计算 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的矩阵形式

$$\boldsymbol{X}=\boldsymbol{Y}^{\mathrm{T}}\times\boldsymbol{P}=\boldsymbol{X}\times(\boldsymbol{P}^{\mathrm{T}}\times\boldsymbol{P}) \tag{7}$$

$$\boldsymbol{Y}^{\mathrm{T}}=\boldsymbol{X}\times\boldsymbol{P}^{\mathrm{T}}=\boldsymbol{Y}^{\mathrm{T}}\times(\boldsymbol{P}\times\boldsymbol{P}^{\mathrm{T}}) \tag{8}$$

由文献[7]可知，对于任意给定的初始向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$，迭代过程都是收敛的. $\boldsymbol{X}$ 将稳定于矩阵 $\boldsymbol{P}^{\mathrm{T}}\times\boldsymbol{P}$ 的某个特征向量上，$\boldsymbol{Y}^{\mathrm{T}}$ 将稳定在 $\boldsymbol{P}\times\boldsymbol{P}^{\mathrm{T}}$ 的某个特征向量上.

由此可以得到如下计算成员引擎 Hub 值向量 $\boldsymbol{Y}$ 和结果 Authority 向量 $\boldsymbol{X}$ 的方法.

算法 1

1) 把查询词提交给元搜索，得到一组返回结果序列；
2) 初始化向量 $\boldsymbol{Y}$；
3) 计算推荐度矩阵 $\boldsymbol{P}$；
4) 重复以下步骤，直至 $\boldsymbol{Y}$ 向量稳定：
   a) 根据式(8)，计算向量 $\boldsymbol{Y}$；
   b) 规范化向量 $\boldsymbol{Y}$；
5) 输出向量 $\boldsymbol{Y}$.

算法 2

1) 把查询词提交给元搜索，得到一组返回结果序列；
2) 初始化向量 $\boldsymbol{X}$；
3) 计算推荐度矩阵 $\boldsymbol{P}$；
4) 重复以下步骤，直至 $\boldsymbol{X}$ 向量稳定：
   a) 根据式(7)计算向量 $\boldsymbol{X}$；
   b) 规范化向量 $\boldsymbol{X}$；
5) 输出向量 $\boldsymbol{X}$.

2.3 成员引擎特征表示

对于不同主题，成员引擎的 Hub 值是不一样的[8-9]. 对某个主题的检索性能较好，Hub 值就较高，对主题的检索性能不好，Hub 值就较低. 另外，成员引擎的 Hub 值只是在 1 次查询过程中对成员引擎的性能评价，这种评价是不稳定的，也不够准确. 因此，需要对 1 组主题关联词查询后得到的 1 组某类主题的 Hub 值进行综合，得到某个成员引擎关于某类主题的主题 Hub 值 S，用来刻画成员引擎在某个主题领域比较稳定的性能评价，计算公式为

$$S=\sum_{k=1}^{L} d_k\times h_k \tag{9}$$

式中，d_k 是第 k 次查询使用的主题关联词的主题关联度；h_k 是第 k 次查询计算得到的 Hub 值；L 是主题关联词的个数. 这里把主题关联词的主题相关度 d_k 作为 Hub 值 h_k 的权重，用来保证与主题关联度高的词查询得到的结果主题相关度高，因此 Hub 值的主题相关度也较高.

综上所述，对于有 m 个成员引擎的元搜索，关于 K 个主题$\{C_1,C_2,\cdots,C_K\}$的成员引擎特征可以表示为特征矩阵

$$\boldsymbol{T}=\begin{bmatrix} t_{1,1} & \cdots & t_{1,K} \\ \vdots & & \vdots \\ t_{m,1} & \cdots & t_{m,K} \end{bmatrix}$$

其中 $t_{i,j}$ 为第 i 个成员引擎在第 j 个主题下的主题 Hub 值.

某个成员搜索引擎的某类主题的主题 Hub 值计算步骤为:

算法 3

1) 产生一组主题关联词;

2) 初始化该类主题的主题 Hub 值;

3) 对于每个主题关联词:

a) 计算主题查询词的主题关联度;

b) 提交查询,计算此次查询的 Hub 值;

c) 对 Hub 值进行加权累加;

4) 输出主题 Hub 值.

2.4 基于主题 Hub 值的结果排序

学习得到了成员引擎的主题 Hub 值,对于一个查询词 t,首先计算 t 属于每个主题的概率 $P(C_1|t)$, $P(C_2|t),\cdots,P(C_K|t)$,设 $P(C_k|t)=\max\{P(C_1|t),P(C_2|t),\cdots,P(C_K|t)\}$,即 t 属于第 k 个主题 C_k 的概率最大,则特征矩阵 $\boldsymbol{T}$ 中的第 k 列向量 $\boldsymbol{Z}$ 为此次查询所需的主题 Hub 向量,然后计算推荐度矩阵 $\boldsymbol{P}$,最后利用

$$\boldsymbol{W}=\boldsymbol{Z}\times\boldsymbol{P} \tag{10}$$

计算排序向量 $\boldsymbol{W}$,按 $\boldsymbol{W}$ 排序结果并返回给用户.

3 实验

3.1 实验数据的构建

实验采用 TanCorp-12 语料集,包含 12 大类,60 小类,共有 14 150 篇文档. 为了保证主题 Hub 值计算的准确性,选择了涵盖面较小的汽车、房地产、人才主题和一个涵盖面较大的计算机主题. 利用文献[10]中的主题关联词提取算法得到每类主题的 200 个主题关联词. 数据表明,在这个语料集上学习得到的主题关联词的主题关联度较好,能保证返回结果以较大的概率属于该类主题.

3.2 成员引擎主题 Hub 值学习

系统目前具有 5 个具有中文检索能力成员引擎,它们分别是谷歌、百度、网易、搜狗和 Live 搜索.

图 2 为汽车主题关联词每次计算得到的成员引擎 Hub 值的部分情况. 纵坐标表示 Hub 值,横坐标表示某次查询. 实验结果表明,成员引擎的 Hub 值表现出较强的稳定性:谷歌和百度的每次查询计算得到的 Hub 值总是高于搜狗、网易和 Live 搜索. 但是在有些查询中,它们的成员 Hub 值表现出波动,这也说明每次查询计算得到的 Hub 值是不稳定的,应该用一组查询才能反映成员引擎某个主题、某段时间的查询性能.

图 3 为系统最终学习到的成员引擎的 4 类主题 Hub 值. 可以看出,成员引擎关于不同主题的 Hub 值具有较大的差异性,并且不同的成员引擎在不同主题上各具优势,没有一个成员引擎在所有主题上都领先. 例如谷歌在汽车主题的 Hub 值高于百度,但是在人力资源主题的 Hub 值却明显低于百度. 成员引擎性能关于主题的差异性为搜索结果按照成员引擎的主题 Hub 值进行排序提供了基础.

3.3 MetaSearch 检索性能

为了对 MetaSearch 检索性能进行评价,将 MetaSearch 和常用的搜索引擎谷歌、网易、百度和优秀的元

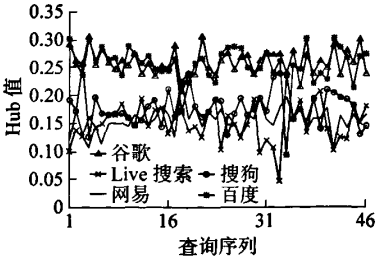

图 2　元搜索每次查询计算的成员引擎 Hub 值

Fig.2　The hub values calculated by index words

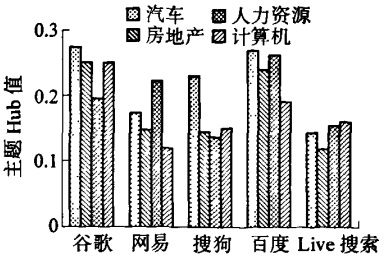

图 3　系统学习得到的 5 个成员引擎的主题 Hub 值

Fig.3　The topic hub values of five search engines

搜索 Vivisimo 进行比较，隐藏 MetaSearch 内部排序机制，把它和这些搜索引擎等同视之．采用在传统信息检索中衡量系统的基本指标：查全率(recall)和查准率(precision)．查全率是检索出的相关文档数和文档集中所有的相关文档数的比率；查准率是检索出的相关文档数与检索出的文档总数的比率．

首先，将查询词依次提交给第 i 个成员引擎，得到结果集 D_i．根据式(7)、(8)计算结果集 $\bigcup_{i=1}^{m} D_i$ 中结果的 Authority 向量，然后按照结果 Authority 向量排序得到排序结果集．最后，取前 k 个结果得到最终结果集 D，D 可以看作查询词的全部相关结果．于是，第 i 个成员引擎的查准率和查全率可以分别定义为

$$\boldsymbol{U}_i = \frac{\|\{r_i \mid r_i \in (D_i \cap D)\}\|}{\| D_i \|} \tag{11}$$

$$\boldsymbol{V}_i = \frac{\|\{r_i \mid r_i \in (D_i \cap D)\}\|}{\| D \|} \tag{12}$$

综合上述 2 个指标得到一个综合指标 F_i 作为成员引擎的评价值

$$F_i = \frac{1 + b^2}{\frac{b^2}{\boldsymbol{V}_i} + \frac{1}{\boldsymbol{U}_i}} \tag{13}$$

其中 b 用于刻画查准率和查全率的相对重要性．

图 4 为 200 次人力资源领域的查询中，搜索引擎谷歌、网易、百度和元搜索 MetaSearch、Vivisimo 的综合指标算术平均值 $\overline{F}$ 的情况．实验数据表明，各个搜索引擎性能之间有较明显的差异，其中采用基于主题 Hub 值的元搜索 MetaSearch 取得了最好的排序效果，比百度高 3%，比元搜索 Vivisimo 高 30%．排第 2 位的是百度，这也印证了百度专注于中文搜索的优势．谷歌和网易的效果不如预期的好，元搜索 Vivisimo 的效果最不好．

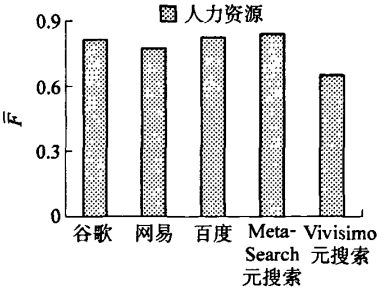

图 4　5 种搜索引擎综合指标的算术平均值 $\overline{F}$

Fig.4　The arithmetic mean $\overline{F}$ for the five search engines

4　结束语

作者利用搜索结果和搜索引擎相互评价的思想，提出计算搜索结果 Authority 值和搜索引擎 Hub 值的算法，用成员引擎的主题 Hub 值作为元搜索中成员引擎的特征表示，并以此为基础进行元搜索的结果排序，该算法获得了更好的排序质量．另外，本文提出对多个引擎进行检索性能评价的指标，为搜索引擎的评价提供了量化的参考．

参考文献：

[1] BRIN S, PAGE L. The anatomy of a large-scale hypertextual web search engine[J]. Computer Networks and ISDN

Systems, 1998, 30(7): 107-117.

[2] 王晓宇, 周傲英. 万维网的链接结构分析及其应用综述[J]. 软件学报, 2003, 14(10): 1768-1780.
WANG Xiao-yu, ZHOU Ao-ying. Linkage analysis for the world wide web and its application: a survey[J]. Journal of Software, 2003, 14(10): 1768-1780. (in Chinese)

[3] MARTIJN Koster. ALIWEB-archie-like indexing in the WEB[J]. Computer Networks and ISDN Systems, 1994, 27(11): 110-120.

[4] LUO Si, CALLAN Jamie. Relevant document distribution estimation method for resource selection[EB/OL]. [2007-05-12]. http://www.cs.cmu.edu/~callan/Papers/sigir03-lsi.pdf.

[5] MENG Wei-yi, YU Clement, LIU King-lup. Building efficient and effective metasearch engines[J]. ACM Computing Surveys, 2002, 34(1): 48-89.

[6] 卜东波, 白硕, 李国杰. 文本聚类中权重计算的对偶性策略[J]. 软件学报, 2002, 13(11): 2083-2089.
BU Dong-bo, BAI Shuo, LI Guo-jie. The duplex strategy of term weighting in text clustering[J]. Journal of Software, 2002, 13(11): 2083-2089. (in Chinese)

[7] DUMAIS S T. LSI meets TREC: a status report[C]// HARMAN D. Proceedings of the 1st Text Retrieval Conference (TREC1). Gaithersburg, Maryland: National Institute of Standards and Technology, 1993: 137-152.

[8] GAUCH Susan, WANG Gui-jun, GOMEZ Mario. Profusion: intelligent fusion from multiple, distributed search engines[J]. Journal of Universal Computing, 1996, 9(2): 637-649.

[9] YUWONO Budi, LEE Dik-lun. Wise: a world wide web resource database system[J]. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(4): 548-554.

[10] 周茜, 赵明生, 扈曼. 中文文本分类中的特征选择研究[J]. 中文信息学报, 2004, 18(3): 18-19.
ZHOU Qian, ZHAO Ming-sheng, HU Man. Study on feature selection in chinesetext categorization[J]. Journal of Chinese Information Processing, 2004, 18(3): 18-19. (in Chinese)

Topic Hub Based Meta Search Engine

JIANG Zong-li, LI Xian-lei, XU Xue-ke
(College of Computer Science, Beijing University of Technology, Beijing 100124, China)

Abstract: To improve the ranking result quality of meta search engine, the authors propose and design a ranking algorithm based on the component engines by topic hub values. The algorithm uses a set of topic relating words to learn the feature of the component engines and denotes them as topic hub values. The ranking algorithm calculates the global similarity of the results according to the topic hub values. The experiments show that the model can get a better ranking quality.

Key words: search engine; meta search engine; ranking; topic hub

(责任编辑 梁 洁)