主题爬虫的设计与实现

林子皓
(南京邮电大学 计算机学院,江苏 南京 210003)

摘 要:在信息化爆炸的时代,一般搜索引擎的搜索结果已经满足不了人们的需要,能获得更准确全面信息的垂直搜索引擎越来越受到关注。其中,主题爬虫作为垂直搜索引擎的核心部分一直是搜索方向的研究热点。文中在分析主题爬虫的结构及特征的基础上,通过引入自己的主题相关度评价方法以及 HITS 网页排序算法,构建了一个主题爬虫。文中给出了爬虫实现的具体步骤,以云计算为主题,进行了实验。实验结果较好地反映了主题爬虫的实用性。

关键词:主题爬虫;HITS 算法;主题相关度
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2014)08-0099-04
doi:10.3969/j.issn.1673-629X.2014.08.023

Design and Implementation of Topic-focused Crawler

LIN Zi-hao
(College of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)

Abstract: In the era of information explosion, the general crawler cannot meet the requirements of personalized search in specific areas, but the topic crawler which can obtain more accurate and comprehensive information gets more attention. Among them, the topic crawler as the core part of the vertical search engine has been the research focus in the search direction. On the basis of analyzing the structure and characteristics of the topic crawler, design a topic crawler by introducing its own measurement of topic similarity and page ranking algorithm of HITS. Offer specific steps of implementing the crawler. An experiment with the theme of cloud computing has been carried out, which proves the practical applicability of topic crawler.

Key words: topic crawler; HITS algorithm; topic similarity

0 引 言

随着信息爆炸式的发展,用户对于信息搜索的需求越来越多。由于一般搜索引擎查询结果广而不精的现状满足不了用户需求,查询更精确、分类更细致、数据更全面的主题搜索引擎应运而生。主题爬虫是主题搜索引擎的关键和基础,它是根据某一特定的主题,在因特网上能自动抓取和主题相关网页的程序。

主题爬虫的主要目标是以特定的方式,高效地抓取 Web 中与主题相关的网页,尽可能过滤与主题无关的链接,实现搜索的专、深、精。它与传统的通用爬虫相比,减少了对资源的利用并且支持扩张性的检索处理。对于主题爬虫而言,最重要的是如何过滤网页中的前向链接,使得爬虫聚焦在一个特定主题的 Web 子集中。

1 主题爬虫模块设计

1.1 整体结构

主题爬虫只爬取与主题相关的网页,并且根据分析、筛选的结果继续爬取合理网页。对比普通爬虫[1-2],需要在原来基础上进行扩充,特别是网页处理部分。整体结构可以看作:初始模块进行初始爬行,主题相关度分析模块进行相关度分析并行适当筛选页面,排序模块对网页的重要性进行一个排序,形成一个优先级序列。每次都从优先级高的网页开始抓取,可以保持主题不偏移。

系统架构图如图 1 所示。

收稿日期:2013-10-28 **修回日期**:2014-01-26 **网络出版时间**:2014-05-21
基金项目:国家自然科学基金资助项目(61170322)
作者简介:林子皓(1988-),男,硕士研究生,研究方向为智能计算技术;导 师:洪 龙,教授,研究员级高级工程师,研究方向为分布式系统、非经典逻辑及应用。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140524.2151.061.html

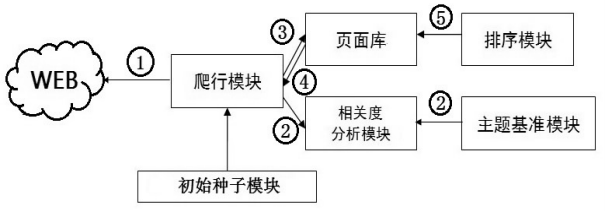

图 1 主题爬虫运行流程及结构示意图

主题爬虫系统运行步骤:

(1)根据爬行模块提供的初始种子及主题,从 Web 中爬取网页;

(2)相关度分析模块对网页进行相关度分析;

(3)由分析结果进行页面的取舍,舍去不符要求的网页;

(4)从后续等待的 URL 中继续取出处理,回到第一步,直到等待为空;

(5)最终排序部分根据自己算法对爬取网页进行重要性排序。

1.2 初始种子和关键词模块

由于主题爬虫是面向特定领域的,具有主体性,初始种子要求是与主题相关领域内的,所以文中定位自主选初始页面,这样能很好地保证主题爬虫从一开始顺利进行下去。例如,所选择的主题是云计算,那么初始种子选择的就是 CSDN 云计算首页。

在确定主题方面,通过对各个关键词赋予相对应的权值,组成相应的关键词集,用这些词集来确定主题。设置权值有人工设置和机器提取两种方法,人工设置即根据以往经验来制定,机器提取就是用程序提取主题网页集中各个网页共同的部分。在文中实现中,为了简便,只统计了种子网页的关键词词频。手工设置通常操作简便,并且设置值与实际情况误差不大,不足之处是有缺漏并且准确性不高;机器提取所定权值更接近标准值,但前提是有主题贴合且代表性和全面性都具备的网页集,否则偏差将大大增加。实际可以结合两种方法,综合它们优点[3]:人工设置关键词赋予权值,搜出相应网页,再用这些网页组成网页集合进行机器提取,得到对应关键词集及权值。

1.3 主题相关度模块

文中把网页的主题相关度作为筛选页面的一个重要的衡量标准,这样做能有效地利用爬虫程序处理掉不相关的网页,避免进行无用爬取,降低准确率。因此必须计算网页主题相关度[4-6],并将相关度小于设定阈值的网页过滤掉。普通爬虫在进行爬取时会对所有 URL 进行处理,没有方向性,这样无疑增大了无效工作量;而主题爬虫会紧扣主题,由主题相关度计算结果来筛选符合网页,去除无用网页,提高准确率,这就是两者的根本区别[7]。

文中将每个关键词看作是一个特征项,作为网页的一个基本单位。通过统计算法 Term Frequency-Inverse Document Frequency(TF-IDF)计算每个特征项的权值。设某一主题在相关页面中特征项的数目为 n,若以 w_j 表示第 j 个特征项在该页面中的权值,则这 n 个特征项权值可以由向量 $\boldsymbol{D}$ 表示:

$$\boldsymbol{D} = (w_1, w_2, \cdots, w_n) \tag{1}$$

定义 1:页面关键词权重频率 $x_i =$ 第 i 个关键词出现的次数 / 页面中关键词出现的最多次数。

分析待判断的页面统计出关键词出现的次数,分别表示为 $a_i(i=1,2,\cdots,n)$,以出现次数最高的关键词作为基准,记为 a_k,则权重频率 $x_i = \frac{a_i}{a_k}$,显然出现最高次数的频率即为 $x_k = \frac{a_k}{a_k} = 1$,这样就能通过页面关键词权重频率反映出第 i 个关键词在页面中的重要程度。

其他关键词的权重频率 x_i 可以根据以上算式算出,那么页面中每一维分向量为 $x_i w_i$,待判定网页的关键词权重向量空间 $\boldsymbol{T}$ 表示为:

$$\boldsymbol{T} = (x_1 w_1, x_2 w_2, \cdots, x_n w_n) \tag{2}$$

文中用主题基准模型向量和待判定网页向量的夹角余弦来衡量其主题相关度。计算公式如下:

$$\mathrm{Sim} = \cos < \boldsymbol{T}, \boldsymbol{D} > = \frac{(\boldsymbol{T}, \boldsymbol{D})}{|\boldsymbol{T}| * |\boldsymbol{D}|} = \frac{x_1 w_1^2 + x_2 w_2^2 + \cdots + x_n w_n^2}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}\sqrt{x_1 w_1^2 + x_2 w_2^2 + \cdots + x_n w_n^2}} \tag{3}$$

当计算的余弦值大于等于系统指定的相关度阈值时,才会认定当前处理页面为主题相关页面。假设阈值为 r,若 $\cos<\boldsymbol{T},\boldsymbol{D}> \geqslant r$,则认为此页面和主题相关,保留并下载此页面;若 $\cos<\boldsymbol{T},\boldsymbol{D}><r$,则认为此页面和主题无关,舍去此页面。$r$ 所设值能决定获得页面的多少,一般根据实际需要来。r 越小,筛选条件低,获得的页面越多;r 越大,筛选条件越严格,获得的页面越少。

1.4 网页价值排序模块

网页价值排序模块是对已筛选留下的网页进行操作,把这些网页按实际价值高低排序,顺序的先后体现了网页的重要程度,也方便价值高的网页容易地被选择到。除了主题相关度因素体现网页重要性以外,其他因素也是所需要排序模块考虑的,诸如网页链接个数、链接指向、被其他网页指向等等[8]。

在 Web 页面中存在大量的超链接,超链接分析可以指出更有价值的搜索方向,可以很好地提高检索质量,HITS 算法是其中一个比较有代表性的算法。HITS

算法是由康奈尔大学(Cornell University)的 JonKleinberg 博士于 1998 年首先提出的[9],HITS 的英文全称为 Hypertext-InducedTopicSearch。

HITS 算法定义了两个重要概念:Authority 页面(某一主题的权威页面)和 Hub 页面(与 Authority 页面连接在一起的页面)。

Authority:代表了特殊领域内或与主题联系紧密的高质量网页。它的权威度与自身提供内容信息有关,即网页被引用的越多,其 Authority 越大,网页越重要。

Hub:代表了一种包含了很多指向高质量页面链接的网页,即提供高质量的超链接。而链接权威度与网页提供的超链接的质量相关,引用内容质量高的网页越多,网页的链接权威度越高。它是一种指向权威网页的链接集合[10]。

重要性排序模块中的网页排序可以将主题相关度和链接分析两个因素结合起来考虑,链接分析部分主要运用上面篇幅所介绍的 HITS 算法[11-12]。以下是排序模块中 HITS 算法流程:

(1)通过主题爬虫获得与主题最相关的 K 个网页的集合,称之为 root 集。

(2)通过连接分析扩展 root 集,扩展后得到的集合称之为 base 集。扩展方法:对于 root 集中任一网页 p,加入所有 p 中所包含的链接到 root 集,加入最多 d 个指向 p 的连接到 base 集。

(3)计算 base 集中所有页面的权威值和中心值:设 n 维向量 $\boldsymbol{a},\boldsymbol{h}$。$a_i,h_i$ 分别表示节点 i 的 Authority 值和 Hub 值。算法如下:初始化向量 $\boldsymbol{a},\boldsymbol{h},a_0=1,h_0=1$,然后进行 I,O 操作。

I 操作:$a_t(v)=\sum_{\langle w,v\rangle\in E}h_{t-1}(w)$

O 操作:$h_t(v)=\sum_{\langle w,v\rangle\in E}a_{t-1}(w)$

(4)规范化。

$$a_t(v)=\frac{a_t(v)}{\sqrt{\sum_{q=0}[a_t(q)]^2}}$$

$$h_t(v)=\frac{h_t(v)}{\sqrt{\sum_{q=0}[h_t(q)]^2}}$$

I 操作反映了如果一个网页有很多好的 Hub 指向,其权威值会相应增加。

O 操作反映了如果一个网页指向很多好的权威页,Hub 值也会相应增加。

重复计算 I,O 操作和规范化,直至 $a(u)$ 和 $h(v)$ 收敛为止。

具体代码实现如下:

a,h 初始化为 1,$a_0=1,h_0=1$

$t=1$

do

for each v in V

do $a_t(v)=\sum_{(w,v)\in E}h_{t-1}(w)$

$h_t(v)=\sum_{(w,v)\in E}a_{t-1}(w)$

$a_t=a_t/\|a_t\|$

$h_t=h_t/\|h_t\|$

$t=t+1$

while $\|a_t-a_{t-1}\|+\|h_t-h_{t-1}\|<\varepsilon$

return (a_t,h_t)

2　主题爬虫具体实现

2.1　URL 相关

实现主题爬虫需要进行主题相关度计算,并根据计算所得值进行页面筛选,文中定于使用 4 个 URL 队列[13-14],各个队列都是同状态 URL 集合:

(1)等待队列:该队列等待程序处理,并且爬虫新爬取的网页将加入其中。

(2)异常队列:无法进行下载的链接将被加入其中,不再进行下一步骤,并且舍弃掉。

(3)抛弃队列:下载可正常进行,但主题相关度小于阈值的链接放进此队列,程序也将不再进行下一步骤。

(4)完成队列:下载可正常进行,但主题相关度大于阈值的链接放进此队列,完成下载后,将已下载过的 URL 加入完成队列。

图 2 说明了 URL 队列的转化流程及 URL 所处各个队列的关系。

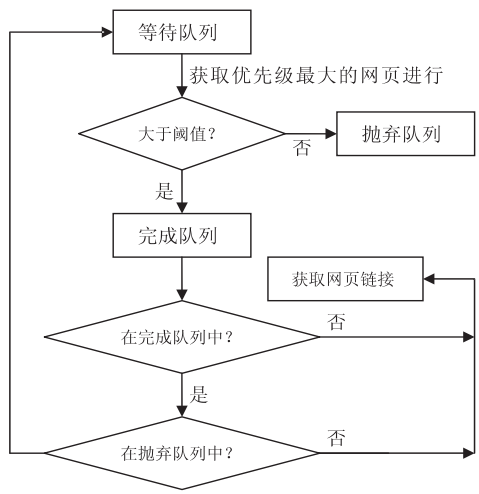

图 2　URL 在队列中的流通过程

2.2　网页爬取

在爬取一个网页之前,首先要检查该网页。如果该网页是一个网络资源,那么就没有必要访问,例如网页是一个 mp3 的下载页。需要忽略的网页类型有:

if (s.endsWith(".zip") || s.endsWith(".gz")

```
|| s.endsWith(".exe") || s.endsWith(".exe")
|| s.endsWith(".jpg") || s.endsWith(".png")
|| s.endsWith(".tar") || s.endsWith(".chm")
|| s.endsWith(".iso") || s.endsWith(".gif")
|| s.endsWith(".csv") || s.endsWith(".pdf")
|| s.endsWith(".doc") || s.endsWith(".rar"))
return false;
else
return true;
```

这些资源文件不值得下载,因为无法解析其内容。

在检查完网页后,去访问网页。如果网页的返回代码不是 403 或者 404 等拒绝或者不存在的值,就可以下载该网页。

为了求相关度,需要统计网页的关键词词频,程序中采用的方法是简单的字符串分割函数。

在本程序中,默认将阈值 r 设置为 0.91。如果一个网页相关度大于阈值,那么就需要提取该网页中的超链接,用到的正则表达式是:<a\s * href = "([^>"] *)"[^>] * >。

3 实验结果分析

文中设计的主题爬虫系统是在真实的局域网环境下运行使用的。由于爬虫需要爬取的信息来自于互联网,因此主题爬虫的运行必须设置在网络畅通的环境下。实验软硬件条件:双核 CPU(Core(TM)2),内存 2 G,操作系统 Win7,Java 语言编写。参数设置:访问总数设为 1 000,线程数 10,阈值 0.91,起始种子 url:http://cloud.csdn.net/。

图 3 为主题爬虫运行结果图;图 4 为主题爬虫相关参数设置。

图 3 主题爬虫运行结果图

用文中设计的主题爬虫和一般爬虫进行了实验对比,结果说明主题爬虫会有效地围绕主题来进行搜索爬取页面,爬取的页面相关性及实用性较好。它会根据相关度丢掉一些无用链接,并且将不再处理,而一般爬虫只会盲目地对所有页面进行处理,这样无论精度还是时间,都会劣于主题爬虫。由实验表明,实现一个只搜集特定领域内信息的主题爬虫是可以完成的,而且主题爬虫的精度还可以调节,阈值的大小控制着搜集页面的精细程度,这样可以灵活地控制想搜集页面的数量。

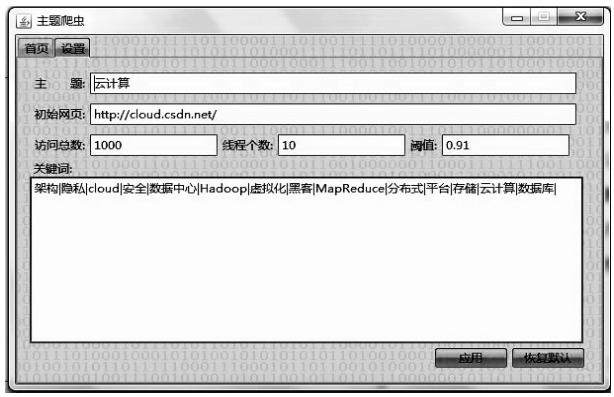

图 4 主题爬虫相关参数设置

4 结束语

文中通过实验理论分析合理地设计了一种主题爬虫,并且用代码做成实例,融合了改进的主题基准模型判断的思想,并且创新地添加了 HITS 算法的排序模块,去除了相对主题无关的网页,并对云计算主题进行了测试,效果较好,相对一般爬虫,搜索的信息和主题更为贴切,为后续做相关的主题搜索引擎打下基础。由于只是单纯的主题爬虫,所以实用性还得配合相应的组合框架来实现垂直搜索引擎的实现,这是后续需要做的工作。

参考文献:

[1] 王凤红. 简单分布式网络爬虫模型的设计与分析[J]. 中国现代教育装备,2008(4):76-78.

[2] 梁云静. 基于遗传算法的主题爬虫搜索策略的研究[D]. 武汉:湖北工业大学,2010.

[3] 汪 涛,樊孝忠. 主题爬虫的设计与实现[J]. 计算机应用,2004,24(S):270-272.

[4] 陈竹敏. 面向垂直搜索引擎的主题爬虫关键技术研究[D]. 济南:山东大学,2008.

[5] Chakrabarti S, van den Berg M, Dom B. Focused crawling: a new approach to topic-specific web resource discovery[J]. Computer Networks, 1999, 31(11-16): 1623-1640.

[6] 邓岳贵. 启发式搜索在网络爬虫中应用的分析[J]. 软件导报,2008,7(2):80-82.

[7] 王 斐. 基于增量反馈和自适应机制的主题爬虫系统的设计与实现[D]. 南京:南京理工大学,2005.

[8] 刘金红,陆余良. 主题网络爬虫研究综述[J]. 计算机应用研究,2007,24(10):26-29.

[9] Kosala R, Blockeel H. Web mining research: a survey[J]. ACM SIGKDD Explorations Newsletter, 2000, 2(1): 1-15.

(下转第 107 页)

图 3 所示。

图 3　扩展后的 Swing 组件

采用同样的扩展方法,同时也实现了 Android 平台的 TextView 和 EditText 组件的扩展。扩展后的组件可以实现蒙古文和中文的同时显示功能以及蒙古文竖排显示和编辑功能。实验结果如图 4 所示。

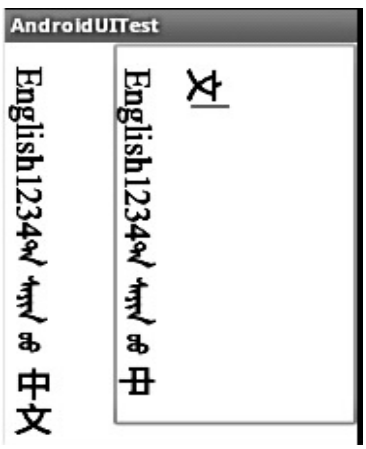

图 4　扩展后的 Android UI 组件

4　结束语

文中对 Swing 组件和 Android UI 组件的蒙古文扩展方法进行了研究,提出了一套扩展方法,并依据此方法对 Swing 的 JLabel、JTextFiled、JTextArea 组件和 Android 的 TextView、EditText 组件进行了实际扩展,实验结果表明扩展方法是稳定有效的。扩展后的组件能够支持蒙古文和中文的同时显示以及蒙古文的竖排显示、编辑,这些组件的实现能够降低基于 Java 的 PC 和 Android 蒙古文软件界面的开发成本,提高软件开发效率,同时使用这些组件还可以将已存在基于 Swing 的软件客户端和 Android 软件界面彻底蒙古文化,满足互联网和移动平台蒙古文应用软件开发的需要。

参考文献:

[1] 周扬荣,贾彦民,吴　健. 基于 ICU 的复杂文本布局引擎设计与跨平台应用研究[J]. 计算机应用研究,2007,24(2):219-221.

[2] 李宗恒,李俭伟. 主要智能手机操作系统发展现状及前景展望[J]. 移动通信,2010(3):115-118.

[3] Loy M,Eckstein R,Wood D,et al. Java Swing[M]. [s. l.]:O'Reilly,2012:13-16.

[4] 公　磊,周　聪. 基于 Android 的移动终端应用程序开发与研究[J]. 计算机与现代化,2008(8):85-89.

[5] 丁二帅. Android 环境下 AppWidget 体系结构研究[D]. 西安:西安电子科技大学,2011.

[6] 罗淑元. Android 系统中 Widget 的设计与实现[D]. 北京:北京交通大学,2012.

[7] 王　乐,周琪云,肖小方,等. Java2 Swing 组件扩充的研究与实现[J]. 计算机与现代化,2010(1):185-187.

[8] 裴　龙,何大可. Java2 Swing 组件设计模式分析[J]. 计算机应用,2001,21(8):274-275.

[9] 戴　歆. Java Swing 程序开发[J]. 软件导刊,2007(9):138-139.

[10] Fowler A. A Swing architecture overview:the inside story on JFC component design[EB/OL]. 1999. http://java. sun. com/products/jfc/tsc/articles/architecture.

[11] Fowler A. Painting in AWT and Swing[R/OL]. 2003. http://www. oracle. com/technetwork/java/painting-140037. html.

[12] ITEEDU. JTextArea 的事件处理[EB/OL]. (2012)[2013]. http://www. iteedu. com/plang/java/jtswingchxshj/49. php.

[13] International Standard ISO/IEC 10646-1 Second Edition. Information technology-Universal multiple Octet coded Character Set(UCS)[S]. 2000.

[14] 确精扎布. 蒙古文编码[M]. 呼和浩特:内蒙古大学出版社,2000.

[15] 巩　政. 蒙古文编码转换研究[D]. 呼和浩特:内蒙古大学,2007.

[16] 姚延栋,吴　健,孙玉芳,等. 传统蒙古文变形显示机制研究与实现[J]. 中文信息学报,2005,19(5):84-89.

(上接第 102 页)

[10] 陈志德,郭扬富,许　力,等. 基于 Hits 算法的 Web 安全改进模型[J]. 武汉大学学报:理学版,2012,58(6):531-534.

[11] 赫枫龄,左万利. 利用超链接信息改进网页爬行器的搜索策略[J]. 吉林大学学报(信息科学版),2005,23(1):59-63.

[12] 贺　晟,程家兴,蔡欣宝. 基于模拟退火算法的主题爬虫[J]. 计算机技术与发展,2009,19(12):55-58.

[13] Aggarwal C C,Al-Garawi F,Yu P S. Intelligent crawling on the World Wide Web with arbitrary predicates[C]//Proceedings of the 10th international WWW conference. Hong Kong:[s. n.],2001:96-105.

[14] Ester M,Groβ M,Kriegel H P. Focused Web crawling:a generic framework for specifying the user interest and for adaptive crawling strategies[C]//Proceedings of 27th international conference on very large data bases. Roma,Italy:[s. n.],2001.

主题爬虫的设计与实现

作者： 林子皓， LIN Zi-hao
作者单位： 南京邮电大学 计算机学院,江苏 南京,210003
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2014(8)