政府网站搜索系统的日志挖掘、行为分析及改进

叶小榕[1],邵晴[2]

1. 中国科学技术信息研究所,北京 100038
2. 北龙中网(北京)科技有限责任公司,北京 100190

摘要 为提高政府网站的搜索质量并优化网站内容,对某政府网站现有搜索系统进行二次开发,增加了日志挖掘模块、行为分析模块、系统改进模块,实现了对搜索系统日志挖掘和用户行为的分析处理。日志挖掘模块负责收集、过滤和识别用户的搜索操作记录;在行为分析模块,根据操作记录从查询过程、聚类分析和查询热词3个角度,分析用户行为的特点和规律,得到了待调整权重的网页和热点查询词等分析结果;在系统改进模块,通过调整网页的权重使查询结果更加精准,改善了搜索系统,根据统计查询热词,既提供了搜索热点等新功能,又为用户提供了个性化网页并优化了政府网站的内容,实现了与舆情系统的数据交互。通过这些优化和改进,从多方面使搜索系统和政府网站能更好的为用户服务。

关键词 政府网站;搜索系统;日志挖掘;行为分析

中图分类号 TP393.09 **文献标志码** A **doi** 10.3981/j.issn.1000-7857.2015.11.017

Log mining, behavioral analysis and improvement of government website search system

YE Xiaorong[1], SHAO Qing[2]

1. Institute of Scientific and Technical Information of China, Beijing 100038, China
2. KNET Co., Ltd., Beijing 100190, China

Abstract In this paper, secondary development was conducted on the search system of one e-government website by adding the log mining module, behavioral analysis module and system improvement module, to improve the search quality and optimize website content. Log mining, processing and analysis of user behaviors have been achieved in the improved search system. The log mining module is able to record, filter and identify the query log. The behavioral analysis module analyzes the characteristics and rules of user behaviors from three aspects including the query process, clustering analysis and hotspot query words, and obtains the results of weights of the webpage and hotspot query words. The system improvement module makes the query results more precise, provides new function of search hotspot and personalized webpage, improves the content of e-government website, and exchanges the data with public opinion system. In this way, the search system and e-government websites will provide users with better service.

Keywords government website; search system; log mining; behavior analysis

目前国内对网站搜索系统优化的研究方法,多以分类模型、聚类分析、关联分析等为主,而结合实际且专门针对政府网站的搜索系统进行日志挖掘、行为分析[1-3]和系统改进等方面的研究还较欠缺。特别是,当前国内越来越重视政府网站的建设,《国家电子政务"十二五"规划》提出"大力推进国家电子政务发展是国家'十二五'的重要任务",强调"强化政府网站应用服务"。而且目前中国政府网站的内容愈加丰富、信息更新速度越来越频繁、栏目的划分也更加细化复杂。据

收稿日期:2014-10-22;修回日期:2015-03-30

作者简介:叶小榕,高级工程师,研究方向为计算机软件、数字图书馆,电子信箱:yeelfine@sina.com

引用格式:叶小榕,邵晴. 政府网站搜索系统的日志挖掘、行为分析及改进[J]. 科技导报, 2015, 33(11): 94-102.

国家信息中心发布的《中国政府网站发展数据报告2012》[4]（以下简称《数据报告》）显示，“中国部委政府网站的页面数量达到了568万个”，用户在访问政府网站时具有明显的目的性，多数是为了快速直接的查询政策信息或办事流程，因此用户会更加依靠政府网站提供的搜索系统。《数据报告》提到，“中国地市级以上政府网站页面搜索可见数达到6652万个，用户总访问量过百亿次。”但搜索系统的查询效率和质量不高，无法高效准确地为用户提供服务，使得政府网站对用户黏性不足。中国软件评测中心、人民网、新浪网、百度共同发布的《2012年中国政府网站绩效评估总报告》[5]指出，“政府网站经常出现搜索结果不准确，甚至搜索系统无法访问等问题。”《数据报告》统计表明，“用户在政府网站的平均停留时间只有2分52秒，网站跳出率高达63.33%。”

针对上述问题，本文通过挖掘和分析政府网站搜索系统的日志记录，对政府网站的搜索系统和内容等实现优化改进，以提高搜索质量、改进政府网站服务效果。

1　系统架构

对某政府网站已有的搜索系统进行二次开发，增加日志挖掘模块、行为分析模块、系统改进模块三大功能模块。

日志挖掘模块[6]，包括日志记录子模块、用户识别和查询识别子模块。其中日志记录模块负责记录用户的原始操作日志，为整个系统提供原始数据；用户识别和查询识别子模块，是对用户日志进行过滤、识别和整理，识别出相互关联的操作，为下一步提供加工好的数据。

行为分析模块，包括查询过程子模块、聚类分析子模块和查询热词子模块，分别从不同的角度对日志挖掘子模块提供的日志数据进行统计、分析，从中发现用户搜索的特征规律。

系统改进模块，利用行为分析模块的分析结果，对搜索系统和政府网站内容加以改进；同时通过舆情交互子模块，将统计分析的数据提交给专门的舆情系统，从而实现舆情信息共享。上述功能模块如图1所示。

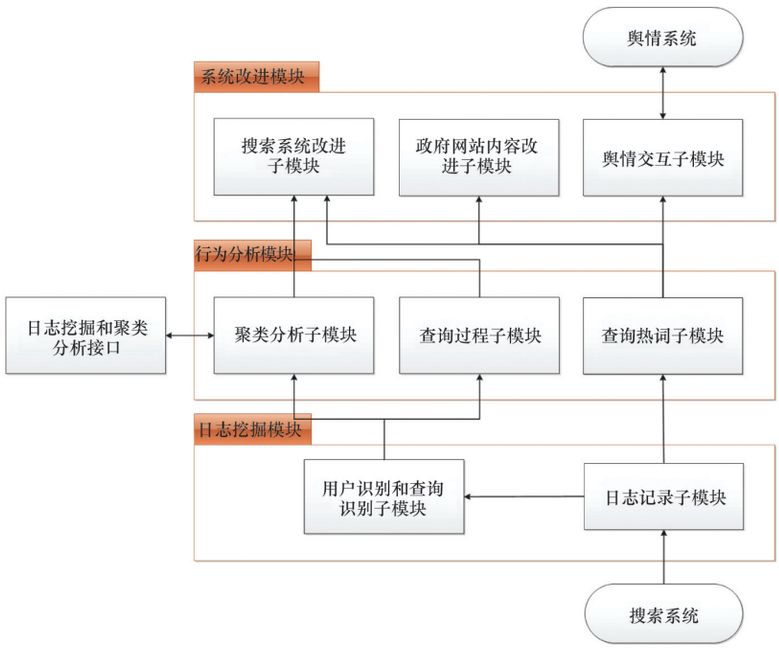

图1　功能模块

Fig. 1　Function module

2　日志挖掘模块

日志挖掘模块负责完整记录使用搜索系统的用户的各项操作，并通过初步的过滤分析，将用户的查询操作过程完整的还原出来，为下一步的行为分析提供数据基础。包括日志记录、用户识别/查询识别2个子模块。

2.1　日志记录子模块

用户行为分析分为显性和隐性两种方式。显性分析要求特定用户主动访问指定的网站，然后通过调查问卷获取分析样本；隐性分析将普通用户的访问日志作为分析数据源，分析结果更加客观，更适合政府网站，因此本系统采用隐性的分析方式。

日志记录子模块采用隐性用户行为分析，在用户查询的每个关键步骤都详细记录用户的原始操作日志，包括查询请求日志query.log、查询结果日志result.log、结果网页打开日志open.log和结果网页操作日志copy.log，如图2所示。

上述日志为日志挖掘提供了原始数据。下面分析各种日志的记录内容。

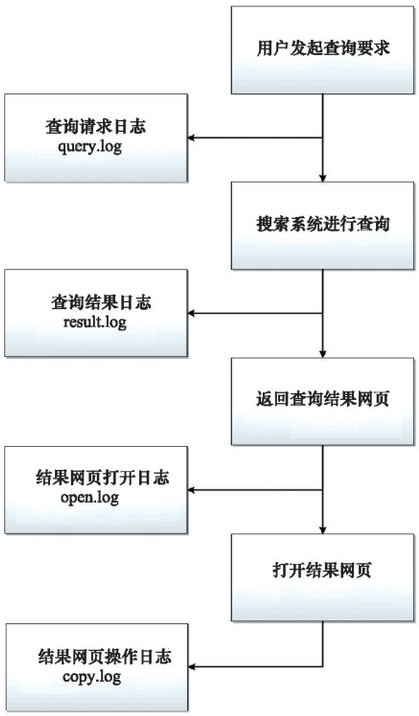

图2 日志记录子模块

Fig. 2 Logging sub-module

2.1.1 查询请求日志

查询请求日志负责记录用户提交给搜索系统的查询请求,一次请求记录为一行。查询日志包括查询时间、当前会话的Session值、查询词QueryWord、浏览器标识UserAgent等,使用#号分割。其中,Session用于区分不同的用户[7];UserAgent分析用户的浏览器;为了将查询请求、查询结果等操作关联起来,每次新的查询请求都生成一个查询唯一键QueryId,且QueryId采用了分布式系统中常见UUID来生成,保证不会出现重复值。查询日志的格式例如:

QueryTime=2013-06-12.10:21:00#Session17F1BB519DF1DB4481A547D04CFE9831#QueryWord=电子政务#UserAgent=Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)#QueryId=83b0b33284fa466c992898a0b4d03a43

2.1.2 查询结果日志

查询结果日志负责将查询结果记录到日志中。包括结果网页的Url地址、结果类型FileType和查询唯一键QueryId等。其中FileType将查询结果区分为网页、文档、图片、视频等不同的类型,便于后文的热门文章统计,格式例如:

Url=http%3A%2F%2Fwww.test.gov.cn%2Fresult%2Ftest1.html&FileType=html&QueryId83b0b33284fa466c992898a0b4d03a43

2.1.3 结果网页打开日志

当用户看到搜索系统返回的查询结果列表后,会点击链接打开自己感兴趣或认为正确的结果网页。这些链接并不直接指向结果网页的URL地址,而是先打开一个跳转地址,将一些有用的信息记录下来,再二次跳转到真正的URL地址,记录和跳转过程非常快速且对用户完全透明,不会影响到用户的使用,跳转链接地址例如:

<a href=”./link?QueryId=83b0b33284fa466c992898a0b4d03a43&PageId1adff5b31a9f47cebb3ea28c90d7d9bd&Order=1&AllCount=30&&Url=http% 3A% 2F% 2Fwww.test.gov.cn% 2Fresult% 2Ftest1.html”/>

其中,PageId为此结果网页的id号,用于跟踪记录此结果网页;Order为此结果网页在总结果集中的序号,从0开始计数;AllCount为总结果集的个数;Url为实际的二次跳转页面地址。

用户点击超链后,搜索系统就会根据上述参数生成结果网页打开日志,格式例如:

OpenTime=2013-06-12.10:25:12#PageId=1adff5b31a9f47cebb3ea28c90d7d9bd#Order=1#AllCount=30##Url=http% 3A%2F%2Fwww.test.gov.cn%2Fresult%2Ftest1.html#QueryId=83b0b33284fa466c992898a0b4d03a43

本日志是最关键的记录,它通过PageId和Url记录了用户认为哪些是正确的结果网页;通过OpenTime记录了打开的顺序;通过Order、AllCount记录了打开的网页在搜索结果集中的排序位置,这些都是日志分析的关键信息。

2.1.4 结果网页操作日志

当用户打开结果网页后,如果用户在此结果网页中有复制、下载、加入收藏夹等操作,表明在此结果网页中,存在用户希望查询的内容,是真正有价值的结果网页。

在网站的所有网页中加入统一的JavaScript脚本,从而当用户有复制等操作时,JavaScript脚本就会往服务器发送Ajax请求,将QueryId、PageId等作为参数发送到搜索系统中记录下来。比如,复制操作的JavaScript代码:

```
document.body.oncopy=function(){//当有复制操作时,就会触发调用此函数
    setTimeout(function(){//复制操100 ms后就发送Ajax请求
        if(clipboardData.getData("text")){//确实进行了复制
            var xmlhttp=new XMLHttpRequest();
            xmlhttp.onreadystatechange=function(){};
            xmlhttp.open("GET","statics?QueryId=17F1BB519DF1DB4481A547D04CFE9831&PageId=1adff5b31a9f47cebb3ea28c90d7d9bd&Operate=copy",true); //拼接发送的报文参数
            xmlhttp.send();//执行发送
    }},100)}
```

其中,参数QueryId和PageId用于跟踪记录结果,Operate表示操作类型为复制。搜索系统接收到后,将复制操作记录为

OperatePageTime=2013-06-12.10:28:12#QueryId=83b0b33284fa466c992898a0b4d03a43#PageId=1adff5b31a9f47cebb3ea28c90d7d9bd#Operate=copy

上述4种日志比较全面地记录了用户查询的关键过程。但搜索系统每天会产生大量的日志,且不同用户的日志会交错记录在一起,这就需要通过用户识别和查询识别子模块进行过滤和筛查,将日志内容分成用户和查询两种维度。

2.2　用户识别/查询识别子模块

对日志进行过滤处理分为用户识别和查询识别[8,9]两步。第一步用户识别,是将每个用户的查询历史分离出来,从而分析单独一个用户历次查询间的关系;第二步查询识别,是在用户识别的基础上,将用户的单次查询再分离出来,方便对用户具体某一次查询的全过程进行分析。识别的处理过程如图3所示。

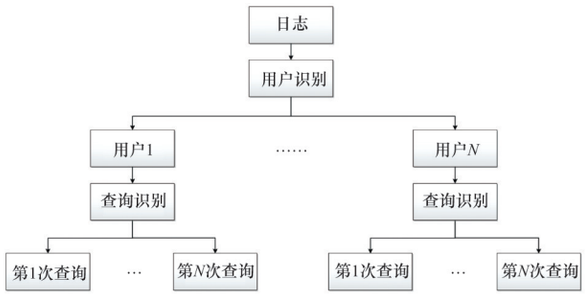

图3　识别处理过程

Fig. 3　Identification process

2.2.1　用户识别子模块

用户识别子模块根据政府网站的用户类型,分为普通用户和登录用户两类,不同用户识别策略也不同。

1）普通用户。普通用户是利用浏览器的Session来做区分。用户第一次访问时创建Session,如果超过30分钟没有操作则Session过期。再次操作时,会创建新Session,系统认定是新用户。

普通用户的识别方法是从查询日志中找到相同的Session值,然后根据Session对应的QueryId值,提取QueryId相同的记录,存入userId.log日志文件中。

2）登陆用户。登录用户是根据用户登录的UserId进行区分,用户查询日志中Session值被UserId值代替,日志格式为:

QueryTime=2013-06-12.10：21：00#Ip=180.153.163.227#UserId=TestUser1#QueryWord=电子政务#UserAgent=Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)#QueryId=83b0b33284fa466c992898a0b4d03a43

登陆用户的识别方法类似普通用户,即从查询日志中找到相同的UserId值,根据UserId对应的QueryId值提取相同的记录,存入userId.log。

普通用户和登录用户的识别方法均需要遍历两次日志,第一次是提取所有的Session和UserId值,第二次将日志中与此Session或UserId相关的记录提取出来,存到userId.log中,比如读取query.log的代码为

```
Set<String> userSet new HashSet<String>();//Session值和UserId值作为键
for(read line in query.log){//循环读取query.log的每一行
    if( line.indexOf("#Session=")>0){//存在Session,说明是普通用户
        String session=getSession(line);//提取Session值
        userSet.add(session);//以Session作为Key
    }else if(line.indexOf("#UserId=")>0){//存在UserId,说明是登陆用户
        String userId=getUserId(line);//提取UserId值
        userSet.add(userId);//以UserId作为Key
}}
for(userId in userSet){//循环遍历userSet
StringBuffer buf=new StringBuffer();//缓存
for(read line in query.log){//循环读取query.log的每一行
    if( line.indexOf(userId)>0){//将存在此UserId的行添加到buf缓存中
        buf.append(line+"\r\n");
}}
save(buf, userId.log);//最后保存到各自的userId.log文件中
}
```

采用同样方式,读取result.log、open.log、copy.log日志,将相同的Session和UserId都存入此userId.log文件中,作为用户识别日志。

2.2.2　查询识别子模块

查询识别子模块在用户识别子模块基础上,从userId.log中将相同QueryId的记录提取出来,从而得到单次的查询记录。代码为

```
Map<String,String>queryIdMap=new HashMap<String,String>();//QueryId作为Key,历次操作记录作为Value
for(read line in userId.log){//循环读取userId.log的每一行
    String queryId=getQueryId(line);//提取QueryId值,作为Key
    String oldLine=queryIdMap.get(queryId);//查询Map中是否已经由此记录
    if(oldLine==null){//如果没有此queryId的记录,就直接放入map中
        querIdMap.put(queryId,line);
    }else{    //如果有记录,则在末尾增加此行记录
        queryIdMap.put(queryId, oldLine+"\r\n"+line);
    }
}
for ( querId in queryIdMap){//循环读取queryIdMap
String lines=queryIdMap.get(queryId);//先得到此QueryId的查询记录
savefile(lines,queryId.log);//最后将查询记录保存到各自
```

的queryId.log文件中

}

这样通过日志挖掘模块对日志进行了过滤处理,将用户查询、打开结果网页、在结果网页上的操作等一系列记录串联起来,保存到用户识别日志userId.log和查询识别日志queryId.log中,为下一步的行为分析模块提供了数据依据。

3 行为分析模块

行为分析模块分为3个子模块。第一个是查询过程子模块,通过读取用户识别日志和查询识别日志,分析用户查询的行为特点和规律特征;第二个聚类分析子模块,利用文献[10]中的聚类分析功能,计算得到网页向量空间模型和用户兴趣模型[11];第三个查询热词子模块,通过计算得出一段时间内的热点查询词。

3.1 查询过程子模块

查询过程子模块是从查询过程的角度,对查询识别日志和用户识别日志进行详细地分析,判断某个结果网页是否符合用户的查询预期,将符合的结果网页增加权重使其排名靠前,反之则降低权重使其排序靠后,使用户能更快地查找到自己需要的内容,最终提高搜索系统质量。

3.1.1 根据查询识别日志设定权重待调整网页

根据查询识别日志中的结果网页打开记录和结果网页操作记录,按如下方式设定待权重调整的网页,在优化搜索结果排序中将进行实际的调整:

1)在结果网页打开记录中,如果用户没有打开排序靠前的网页,而打开了排名靠后的网页,比如存在日志记录为OpenTime=2013-06-12.10:25:12.301#...#Order=2的记录,但是没有Order=0和1的记录,表明用户只打开了排名第3的网页,而Order=0和1这两个原本排名靠前的网页不符合用户预期,就将降低其权重。为简化计算,本文只降低日志记录中比最小Order序号还小的网页权重,其他情况暂不考虑。

2)在结果网页操作记录中存在某个网页的记录,比如日志记录为OperatePageTime=2013-06-12.10:28:12#...PageId=1adff5b31a9f47cebb3ea28c90d7d9bd#Operate=copy,则说明此结果网页是用户需要的内容,将提高其权重。

上述2种方式进行权重调整的代码为

```
List<String>operatePageIdList=new ArrayList<String>();//记录有操作的PageId集合
List<String>unopenPageIdList=new ArrayList<String>();//记录未打开的PageId集合
int minOrder=Integer.MAX_VALUE;//记录最小编号的打开网页
for (read line in queryId.log){//循环读取queryId.log的每一行
 if (line.indexOf("OpenTime=")>=0){//说明此条记录是结果网页打开记录
  int order=getOrder(line);//从此行中得到Order值
  minOrder=(minOrder>order)?order:minOrder;
 }
 if( line.indexOf("OperatePageTime=")>=0){//说明此条记录是结果网页操作记录
  String pageId=getPageId(line);//得到此网页的PageId;
  operatePageIdList.add(pageId);//添加到结果集中
}}
for(int orderIndex=0;orderIndex<minOrder;orderIndex++){//循环最小打开网页前面的网页
 String pageId=getPageId(queryId.log, orderIndex);//从userId.log文件排名比minOrder低但未打开的网页PageId取出来
 unopenPageIdList.add(pageId);//添加到结果集中
}
```

最终,得到了需要降低权重的未打开网页集合unopenPageIdList,及需要提升权重的操作网页集合operatePageIdList,均输入到系统改进模块中,从而进行权重调整。

3.1.2 根据用户识别日志设定权重待调整网页

从用户识别日志userId.log中得到用户历次的检索词,并利用系统已有的相似词词典库,判断用户此次的检索词和以前历次查询的检索词是否相似[12],来设定待权重调整的网页,流程见图4。

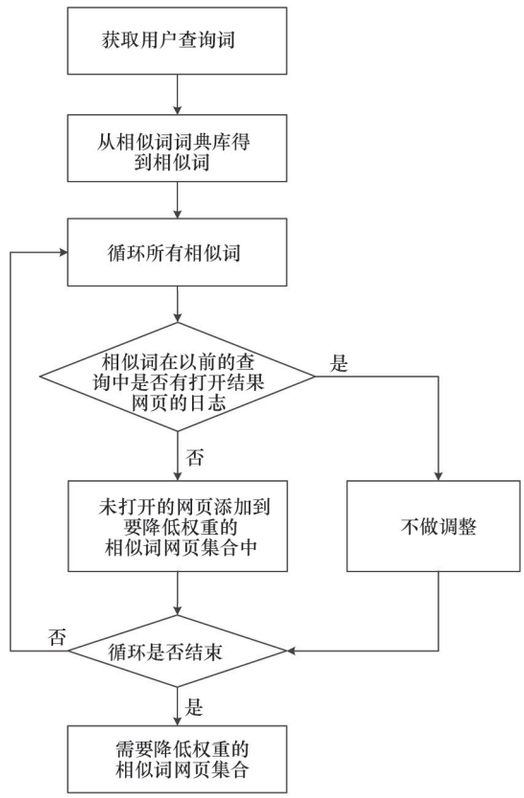

图4 根据用户识别日志进行权重调整流程

Fig. 4 Weight adjustment flowchart based on user identification log

1）当用户识别日志 userId.log 中没有相似词时，说明用户历次查询之间没有关联，不需要调整。

2）当用户识别日志 userId.log 中存在相似词时，说明用户查询之间有关联关系：如果前次查询打开过结果网页，则不进行权重调整；如果前次查询没打开过任何一个结果网页，而只在本次查询中打开过结果网页，则认为用户对前次查询的结果不满意，将降低前面历次结果网页的权重，代码为

```
List<String>similarPageIdList=new ArrayList<String>();//要降低权重的相似词网页集合
Set<String>queryWordSet=read(userId.log);//从用户识别日志中得到历次查询词
for ( String queryWord:queryWordSet){ //循环所有查询词
    Set<String>similarWords=getSimliarWords(queryWord);//得到查询词的相似词集合
    for(String similarWord:similarWords) {
        boolean opened=getOpend(userId.log,similarWord);//根据相似词查询以前的查询结果是否打开过；
        if(!opened){ //如果未打开过
            similarPageIdList.add(getUnOpenPageId(userId.log,similarWord));//将未打开的网页添加到集合中
}}}
```

将得到的需要降低权重的相似词网页集合 similarPageIdList，输入到系统改进模块中，从而进行权重调整。

3.2　聚类分析子模块

此模块调用文献[10]开发的日志挖掘和聚类分析接口，将用户日志 userId.log 作为输入，利用 MapReduce 和 K-means 算法计算得到网页向量空间模型和用户兴趣模型，将兴趣相似的用户组成一个兴趣组，将其结果 Set<UserId,Ui>返回给本模块，其中 Ui 为用户 UserId 从属的用户兴趣模型，对于用户感兴趣的网页集合 clusterPageIdList，输入到系统改进模块中，将其权重提前。因此，通过调用外部接口，也能对网页进行权重调整。

3.3　查询热词子模块

查询热词子模块通过直接分析查询请求日志和查询结果日志，统计出一定时间段内用户的查询热点词和热门文章。

考虑到日志数据量大，因此查询热词功能采用 Hive 实现。Hive 是一个基于 Hadoop 的开源数据仓库工具，用于存储和处理海量结构化数据。处理日志的步骤是创建表、导入数据、使用类似 Sql 的 HiveSql 语句查询所需要的数据。Hive 优点是支持大数据量的批量处理，不足是查询速度较慢、不支持实时查询。为提高查询速度，本系统每晚定期处理日志数据，并将 Hive 统计得到的查询热词结果保存到分布式缓存 Redis 中，前台查询时将直接从 Redis 中获取查询热词。Redis 是一个高性能、基于内存 Key-Value 数据库。

3.3.1　创建数据表

Hive 数据表需与日志中的字段顺序和类型一一对应。首先根据查询请求日志 query.log 创建查询请求表 query.db，数据表的设计见表 1。

表 1　查询请求

Table 1　Query request database

列名	类型	说明
QUERY_TIME	TIMESTAMP	查询时间
SESSION	STRING	查询时的 Session 值
QUERY_WORD	STRING	查询词
USER_AGENT	STRING	浏览器类型
QUERY_ID	STRING	查询 QueryId 值

根据表格的定义，创建表的 Sql 语句为

```
CREATE TABLE QUERY (QUERY_TIME TIMESTAMP, SESSION STRING, QUERY_WORD STRING,USER_AGENT STRING,QUERY_ID STRING)//创建表的字段名和类型
    ROW FORMAT DELIMITED  //声明分隔格式
    FIELDS TERMINATED BY '#' //声明各字段间用#作为分隔符
    LINES TERMINATED BY '\n' //声明每行间用\n作为分隔符
    STORED AS TEXTFILE;//声明按文本文件方式保存，不压缩
```

使用以上语句通过 Hive 控制台创建表成功后，HDFS 中就会创建对应的数据库文件，路径在 hdfs://hadoopserver/user/hive/warehouse/query.db。

以同样的方式，将各个日志记录分别创建各自的数据表，其中比较关键的表为查询结果表 result.db、结果网页打开表 open.db、结果网页操作表 copy.db、用户识别表 userId.db 和查询识别表 queryId.db。

3.3.2　导入数据

Hive 中的数据通过加载文本文件的方式将数据加载到表中，文件中的字段和数据表中的列一一对应。系统每天定期执行脚本，将前 1 天的日志数据导入到 Hive 中。

```
hive>LOAD DATA LOCAL INPATH 'query.log' INTO TABLE query.db;//将本地日志文件 query.log 导入 query.db 表中。
```

以同样的方式，每天定期将其他几种日志文件导入对应的 Hive 表中。

3.3.3　Hive 查询

Hive 提供了丰富的操作接口，系统通过接口每天定时统计出前一天的查询热词，来向 Redis 提供缓存数据，代码为

```
Class.forName("org.apache.hadoop.hive.jdbc.HiveDriver");//加载 Hive 驱动
Connection  con=DriverManager.getConnection("jdbc:hive://
```

```
hiveserver:10000/default", "", "");//建立与数据库的连接
String querySql="select QUERY_WORD,count (QUERY_WORD) as num from query"//从query表查询,返回值包括热词和热词出现的次数
+"where QUERY_TIME>=to_date(from_unixtime(unix_timestamp()- 1*60*60*24, 'yyyyMMdd')) and QUERY_TIME<to_date(from_unixtime(unix_timestamp(), 'yyyyMMdd'))"//时间段位大于等于昨天,并小于今天
+"group by QUERY_WORD"//按查询词分组
+"order by count desc"//按出现次数从大到小排序
+"limit 10";//取前10位
Statement stmt=con.createStatement();//建立到Hive的连接
ResultSet res=stmt.executeQuery(querySql);//执行查询语句
Map<String, Long> hotMap=new LinkedHashMap<String, Long>();//保存热词和出现次数,并按顺序保存
while (res.next()){
    String qw=res.getString(1);//热词
    Long count=res.getLong(2);//热词出现的次数
    hotMap.put(qw, count);//查询结果保存到hotMap中
}
saveQueryHot(hotMap);//结果刷新到Redis缓存中
stmt.close();con.close();//关闭连接
```

通过修改上面代码中的查询条件,可以查询上1周、上1个月的搜索热词,查询result.db库得到热门的查询结果网页,查询userId.db库得到用户的查询热词和查询历史。

3.3.4 存入和查询Redis缓存

为解决Hive查询缓慢的问题,本系统使用Redis提供缓存服务。系统将Hive查询到的结果,更新到Redis中,提供高速的缓存查询服务。这样既利用了Hive的大数据处理功能,又充分发挥了Redis高速缓存的查询功能。更新Redis的代码为

```
public void saveQueryHot(LinkedHashMap<String,Long>hotMap){
    JedisPool pool=new JedisPool(RedisServerIP, port);//连接Redis服务器
    Jedis jedis=pool.getResource();  //获取Redis连接
    jedis.hmset("queryHot", hotMap);  //将Hive的查询结果保存到Redis中
}
```

进行查询时,系统就直接读取Redis中的缓存,迅速得到结果,代码如下:

```
public void getQueryHot(){
  LinkedHashMap<String,Long>hotMap=(LinkedHashMap<String, Long>)redis.hgetAll("queryHot");//从Redis中取出Hive的查询结果hotMap
   for(Entry<String,Long> entry: map.entrySet()) {//循环hotMap仍然是按次数顺序输出
    System.out.print("查询热词:"+entry.getKey()+"出现次数:"+entry.getValue());
}}
```

同样的方式,通过使用Hive查询query.db,并把结果更新到Redis中,就可以得到用户查询热词和查询历史。这些都将用来改进搜索系统和政府网站内容。

4 系统改进模块

系统改进模块是利用行为分析模块提供的分析结果,优化和改进搜索系统和政府网站,并实现与舆情系统的信息交互。

4.1 搜索系统改进

以应用Solr软件作为搜索系统的政府网站为例进行改进,Solr软件是开源、安全、高效且支持多种文档的全文搜索系统[13,14],已被whitehouse.gov、aol.com、nasa.gov、ebay.com等知名大型网站所采用。本模块根据行为分析模块提供的分析结果,优化搜索结果排序、显示搜索热点,从而提高搜索效果、改进政府网站服务效果,增强对用户的黏性。

4.1.1 优化搜索结果排序

此部分根据查询过程子模块和聚类分析子模块中的分析结果来对网页进行权重调整,从而优化查询结果的排序[15]。权重调整是通过Solr软件提供的接口,按照Lucene的评分机制[16,17],修改网页的权重值来实现。

Lucence的评分机制综合使用了信息检索的向量空间模型和布尔模型,通过对每个网页综合打分确定其权重,从而决定此网页的排序位置。因此,通过提高或者降低网页的权重,就可改变查询的排序顺序。Lucene的评分机制为

$$score(q,d)=coord(q,d)\cdot queryNorm(q)\cdot \sum_{t\ in\ d}\left(tf(t\ in\ d)\cdot idf(t)^2\cdot t.getBoost()\cdot norm(t,d)\right) \quad (1)$$

最后1个参数norm(t,d),由公式(2)决定:

$$norm(t,d)=doc.getBoost()\cdot length\ Norm(field)\cdot \prod_{field\ f\ in\ d\ named\ as\ t} f.getBoost() \quad (2)$$

式(2)通过doc.getBoost()确定某个文档的加权分数(称为Document boost),分数默认为1.0f,分数越高排名越靠前。通过修改此加权分数,就可以调整搜索结果网页的排名权重。比如,针对未打开网页集合unopenPageIdList,利用Solr提供的接口进行权重调整的代码如下:

```
for(String pageId in unopenPageIdList){//循环遍历需要降低权重的数组unopenPageIdList
File page=getFile(pageId);//根据pageId得到文件属性
SolrInputDocument doc1=new SolrInputDocument();//创建Solr对象
doc1.addField("title", page.getTitle());//添加网页标题
doc1.addField("url", page.getUrl());//添加网页url
//…..添加其他字段
doc1.setDocumentBoost(0.9f);//需要降低加权分数,即降低
```

权重

SolrServer.add(doc1);//添加到Solr服务器中

}

SolrServer.commit();//循环完毕后,统一提交

通过函数setDocumentBoost(),将此网页的加权分数降低为0.9f。采用同样的方法,遍历operatePageIdList和similarPageIdList,通过升高或降低其权重,使其网页的排名提前或靠后,实现对网页的排序调整。

4.1.2　显示搜索热点

在搜索页面提供“热点查询词”、“热门文章”和“搜索推荐”功能。

“热点查询词”和“热门文章”提供了网站排名前10位的热点查询词和热门查询文章,是应用查询热词子模块,从Redis中的查询热词得到,搜索页面每天均更新显示。对于登陆用户,系统根据此用户的搜索记录,显示一段时间内用户自己的搜索热词和搜索历史。

“搜索推荐”是根据用户以前的搜索情况来提供其可能感兴趣的内容。对于登陆用户,也是从Redis中得到用户以前的搜索历史,进行推荐。网页截图见图5。

图5　搜索页面

Fig. 5　Search start page

4.2　政府网站内容改进

根据查询热词子模块的分析结果,可用来改善政府网站的内容。

4.2.1　优化页面显示

通过Redis中保存的热点词,能够发现大量用户当前关注的内容。比如很多用户频繁搜索某个办事流程,表明此办事流程在网站中没有突出显示,用户不易找到。据此可对政府网站的内容和栏目进行改进,将用户频繁搜索的内容和栏目在网站更明显的位置展示。

4.2.2　个性化页面和消息推送

Redis中保存了每个用户查询热词的历史记录,当有新的信息内容网页发布时,通过检索页面中是否存在此用户查询过的热词,从而实现个性化页面展示和消息推送,代码如下:

```
public void getUserPageAndMsg(String userId){
    LinkedHashMap<String,Long>hotMap=
(LinkedHashMap<String,Long>)redis.hgetAll("userHot"+
userId);//从Redis中取出根据UserId取出此用户的查询热词
List<Page>userPage=new ArrayList<Page>();//保存个性化
页面
List<String>userMsg=new ArrayList<String>();//保存需要
推送的消息
for(Entry<String,Long>entry: map.entrySet()){//遍历查询过
的热词
    String userHot=entry.getKey();//查询的热词为Key
    for(Page newpage in newPageList){//遍历新的网页
        if(newPage.indexOf(userHot)>=0){//如果新网页中
包含热词
            userPage.add(newPage);//将放到个性化页面中
            userMsg.add(newPage.title);//将网页的标题推送
给用户
}}}}
```

通过以上代码就能显示个性化页面,并将包含用户关注内容的页面及时推送给用户。

4.3　与舆情系统交互

利用查询热词子模块,搜索系统实现了与舆情系统的交互。此部分有两个功能:一方面,从舆情系统中获取需要关注的重点词汇,定时统计这些词汇的查询量和排名,将其变化趋势上报给舆情系统,供决策分析使用;另一方面,每天定时将热点词汇上报给舆情系统,比如当有新政策发布时,相关词汇的查询排名会迅速提高,表明了此新政策获得了大量关注,成为了舆论的新热点[18]。其交互关系如图6所示。

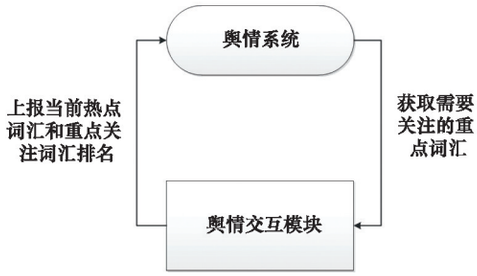

图6 舆情交互关系

Fig. 6 Interaction diagram of public opinions

5 结论

通过对日志挖掘模块、行为分析模块和系统改进模块的开发,基本实现了对用户搜索行为的记录、分析和对现有系统的改进。系统已部署到某政府网站进行试运行,硬件资源由5台试验机组成,cpu分别为8核到16核的志强处理器,内存为8 G到24 G;软件包括HDFS的Hadoop-2.2.0,Hive-0.12.0和Redis-2.8.4。在试运行期间,每天1:00起执行定时任务,分析处理前1天的日志。在搜索系统优化方面,每天约有近百个网页的权重会进行调整,初期调整比较大,后期逐渐稳定;搜索结果排序也得到了优化,用户越来越倾向于打开排名靠前的网页,排名前5的打开网页数占了总打开数的80%左右;新增的搜索热词排行榜及时地反应了前1天的关注热点,用户点击量稳步上升,已得到了用户的认可。在政府网站的内容优化方面,已累计优化修改了数十处的网页布局,更加突出当前的热点;对数万登录用户,都会显示自己的搜索热词和搜索历史,并且定期更新其个性化页面,及时推送有相关内容的更新网页,使用户驻留时间增加了数倍。新增的与舆情系统的交互功能,也更便于掌握舆情动态。

综上所述,通过挖掘搜索系统日志,分析用户行为,可优化搜索系统、改善政府网站的内容、提高政府网站的用户体验,使其能更好地为广大用户服务。在此基础上,还需要进一步完善,比如需要实现分布式多节点的日志汇总功能。

参考文献(References)

[1] 詹圣君. 基于用户行为日志分析的搜索引擎排序算法研究[D]. 武汉: 湖北工业大学, 2011.
Zhan Shengjun. Based on user behavior log analysis of search engine ranking algorithm[D]. Wuhan: Hubei University of Technology, 2011.

[2] 岑荣伟, 刘奕群, 张敏. 基于日志挖掘的搜索引擎用户行为分析[J]. 中文信息学报, 2010, 24(3): 49-54.
Ceng Rongwei, Liu Yiqun, Zhang Min. Search engine user behavior analysis based on log mining[J]. Journal of Chinese Information Processing, 2010, 24(3): 49-54.

[3] 刘承启, 邓庚盛, 江婕. 基于用户行为分析的搜索引擎研究[J]. 计算机与现代化, 2008(9): 75-77.
Liu Chengqi, Deng Gengsheng, Jiang jie. Research on search engine based on user behavior analysis[J]. Computer and Modernization, 2008 (9): 75-77.

[4] 国家信息中心网络政府研究中心. 中国政府网站发展数据报告(2012)[EB/OL]. (2012-12-06) [2013-09-01]. http://www.gwd.gov.cn/uploads/worddownload/2012_development_report_of_governments´_website.pdf.
E-government Research Center of State Information Center. Development data report of Chinese government website(2012)[EB/OL]. (2012-12-06) [2013-09-01]. http://www.gwd.gov.cn/uploads/worddownload/2012_development_report_of_governments´_website.pdf.

[5] 中国软件测评中心. 2012年中国政府网站绩效评估总报告[EB/OL]. (2012-12-05) [2013-09-01]. http://www.cstc.org.cn/zhuanti/fbh2012/zbg1/zbg.html.
China Software Testing Center. The general report of Chinese government website performance evaluation in 2012[EB/OL]. (2012-12-05) [2013-09-01]. http://www.cstc.org.cn/zhuanti/fbh2012/zbg1/zbg.html.

[6] 陈红涛, 杨放春, 陈磊. 基于大规模中文搜索引擎的搜索日志挖掘[J]. 计算机应用研究, 2008(6): 1663-1665.
Chen Hongtao, Yang Fangchun, Chen Lei. Mining query log of large-scale Chinese search engine[J]. Application Research of Computers, 2008(6): 1663-1665.

[7] 张磊, 李亚楠, 王斌. 网页搜索引擎查询日志的Session划分研究[J]. 中文信息学报, 2009, 23(2): 54-61.
Zhan Lei, Li Yanan, Wang Bin. Session segmentation based on query logs of web search[J]. Journal of Chinese Information Processing, 2009, 23(2): 54-61.

[8] Heasoo H, Hady W L, Lise G, et al. Organizing user search histories[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(5): 912-925.

[9] 邱娣. 基于Web日志挖掘的用户信息需求识别研究[D]. 武汉: 华中师范大学, 2012.
Qiu Di. Research on user information demand of recognition based on web log mining[D]. Wuhan: Central China Normal University, 2012.

[10] 叶小榕, 邵晴. 政府网站移动搜索的日志挖掘和个性化改进[J]. 科技导报, 2014, 32(36): 110-116.
Ye Xiaorong, Shao Qing. Log mining and personalization improvements for mobile search system of government websites[J]. Science & Technology Review, 2014, 32(36): 110-116.

[11] Qian Xueming, Feng He, Zhao Guoshuai, et al. Personalized recommendation combining user interest and social circle[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 26(7): 1763-1777.

[12] 宋宇轩. 基于搜索日志和点击日志的同义词挖掘的研究和实现[D]. 北京: 北京交通大学, 2011.
Song Yuxuan. The research and implementation of synonyms mining method based on the search log and click log[D]. Beijing: Beijing Jiaotong University, 2011.

[13] 乐嘉锦, 姚岚. 基于Solr的体育视频信息全文搜索研究[J]. 计算机工程, 2012, 38(24): 269-273.
Le Jiajin, Yao Lan. Research on full-text search of sports video information based on Solr[J]. Computer Engineering, 2012, 38(24): 269-273.

[14] The Apache Software Foundation. Public websites using Solr[EB/OL]. (2013-09-19) [2013-10-01]. http://wiki.apache.org/solr/PublicServers.

[15] Yadav D, Sonia S C, Jorge M, et al. An approach for spatial search using Solr[C]//Confluence 2013: The Next Generation Information Technology Summit (4th International Conference). Noida, India: IET, 2013: 202-208.

[16] 闻峥. 基于Lucene的搜索引擎优化[D]. 北京: 北京交通大学, 2011.
Wen Zheng. Search engine optimization based on lucene[D]. Beijing: Beijing Jiaotong University, 2011.

[17] Saravanakumar K, Aswani K C. Optimized web search results through additional retrieval lists inferred using wordnet similarity measure[C]// International Conference on Data Mining and Intelligent Computing 2014. New Delhi, India: IEEE Conference Publications, 2014: 1-7.

[18] 王宏勇. 网络舆情热点发现与分析研究[D]. 成都: 西南交通大学, 2011.
Wang Hongyong. Hot-topic detection and analysis on internet public opinion[D]. Chengdu: Southwest Jiaotong University, 2011.

(责任编辑 陈广仁)