政府网站移动搜索的日志挖掘和个性化改进

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 为充分利用移动搜索和政府网站的特点,发挥Hadoop处理大数据的优势,设计开发了日志挖掘和个性化定制系统。利用Flume和HDFS实现了海量日志的汇总和存储,为日志挖掘提供了数据源和调用接口;采用MapReduce实现了对日志的高效分析,利用搜索结果网页的标签和导航,建立了网页向量空间模型和用户兴趣模型;根据用户兴趣模型,使用聚类分析中的*K*-means算法将有相似兴趣的用户组成兴趣组;通过计算搜索结果网页到用户所在兴趣组的距离,判断用户对该网页是否感兴趣,据此调整搜索结果的排序,实现个性化搜索和推送功能。

关键词 个性化搜索;个性化推荐;聚类分析;MapReduce

中图分类号 TP393.09 **文献标志码** A **doi** 10.3981/j.issn.1000-7857.2014.36.018

Log Mining and Personalization Improvement for Mobile Search System of Government Websites

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 By taking full advantage of the characteristics of mobile search and government website, a log mining and customization system, which makes use of the advantages of Hadoop in large data processing, is designed and developed. First, it uses Flume and HDFS to realize the collection and storage of massive log and to provide source data and program interface of log mining. Second, the system uses MapReduce to efficiently analyze the log by taking advantage of labels and navigation bar of search result pages. Thus, the vector space model of search result pages and user interest model are established. Third, based on user interest model and combined with MapReduce again, the *K*-means algorithm which is for cluster analysis is used. Then, users are divided into different interest groups depending on their interests. Finally, by calculating the distance between search result page and the user's interest group, whether the user is interested in this page is determined, then the system adjusts the order of search results and pushes a new page to this user accordingly. Therefore, the personalized search and push function are implemented.

Keywords personalized search; personalized recommendations; cluster analysis; MapReduce

随着移动通信技术的迅速发展,越来越多的用户使用手机进行移动搜索。移动搜索是搜索引擎和移动互联网相结合的产物,指用户使用移动终端设备发起的搜索,包括网页搜索、App搜索、微信搜索等。根据中国互联网络信息中心(CNNIC)2014年7月21日发布的《第34次中国互联网络发展状况统计报告》[1],目前中国移动搜索用户数达4.06亿,使用率达到77.0%,移动搜索成为除即时通信外的第二大手机应用。

与传统计算机搜索不同,移动搜索有如下特点[2,3]:1)移动搜索更能完整记录用户的每个搜索行为,系统利用用户的

收稿日期:2014-10-22;修回日期:2014-11-20

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.

登陆信息能精确完整的记录用户的搜索历史、习惯和兴趣；2）移动搜索的查询词更短；3）移动搜索的用户通常关注排名靠前的搜索结果；4）从App客户端到微信都有成熟的信息推送机制可供移动搜索使用。

目前政府网站的移动搜索，主要是为政府网站的网页、App和微信提供移动搜索服务。虽然为用户提供了便捷的搜索服务，但尚存在不少缺陷，例如功能单一，仅提供简单的搜索功能，没有充分利用上文中提到的移动搜索的特点，也没有利用政府网站网页标签精确、导航规范的优势；未对记录的日志数据进行充分挖掘，查询效果较低，也没有将符合用户兴趣的网页排序靠前，无法提供个性化服务，缺少主动推送服务；海量的日志文件分散在各个分布式节点上，缺乏有效的技术手段进行高效的汇总和分析挖掘。

针对以上不足，本研究组设计开发政府网站移动搜索的日志挖掘和个性化定制系统，利用Hadoop处理大数据的优势[4]，通过Flume和HDFS解决海量日志的收集、汇总和存储；采用聚类分析挖掘日志中的用户行为，应用MapReduce实现*K*-means聚类算法，进行量化计算和分析，建立网页的向量空间模型和用户兴趣模型；利用聚类分析的结果为用户供个性化的搜索和推送服务。

1　系统架构

本系统着重利用政府网站和移动搜索的特点，分析移动搜索的日志，特别是用户在搜索结果中打开的网页地址记录，实现日志挖掘和个性化改进。

系统包括日志数据处理、日志挖掘和聚类分析、个性化搜索和推送3大模块[5,6]。日志数据处理模块，负责提供基础数据，从各个分布式节点采集用户搜索的原始日志，进行清洗、统一格式，利用Flume将日志上传并存储到HDFS中；日志挖掘和聚类分析模块，利用MapReduce实现 *K*-means聚类算法，计算出网页的向量空间模型，建立用户的兴趣模型和兴趣组；在以上基础上，通过个性化搜索和推送模块，将用户感兴趣的搜索结果网页排序提前，并主动推送给用户，从而提高移动搜索的查询质量，增强系统与用户的互动，增大系统对用户的黏性。系统整体架构如图1所示。

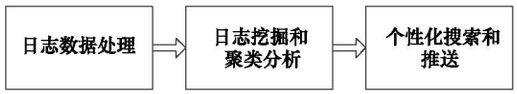

图1　系统整体架构

Fig. 1　Overall system architecture diagram

2　日志数据处理

日志数据处理模块包括日志预处理、日志汇总、日志采集和存储3个阶段。日志预处理，是在系统的各个分布式节点上，对日志内容进行数据清洗、用户识别和会话识别、路径补充和格式统一；日志汇总，是在各个分布式节点上通过Flume将日志汇总上传；日志采集和存储，负责将Flume提交的日志存储到HDFS中，为下一步日志挖掘提供数据源和调用接口。处理流程如图2所示。

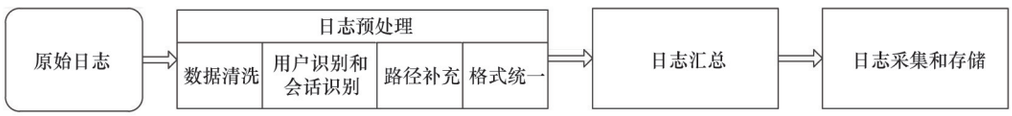

图2　日志数据处理流程

Fig.2　Log data processing flow chart

2.1　日志预处理

日志预处理包括数据清洗、用户识别和会话识别、路径补充和格式统一等。数据清洗负责去除原始日志中冗余、错误、异常和无效的数据；用户识别和会话识别，利用移动搜索的特性，根据移动网页、App和微信中登陆的用户名、手机号等信息，得到用户的UserId，从而将用户做精确的区分；路径补充和格式统一，负责补充上下文信息，使日志记录的每一行都足够完整，并将移动网页、App和微信中的不同日志格式转化为统一的记录格式。

通过日志预处理后的日志大小会降为原来的1/3，减轻存储的压力，便于下一步的数据挖掘。通过此阶段处理后的日志包括用户的UserId、用户打开的搜索结果网页Url地址和网页的标签和导航关键字Label，以#号分割：

UserId=User1#Url=http% 3A% 2F% 2Fwww.***.gov.cn% 2Ftest1.html#Label=政策;监管;电子政务；***条例。

2.2　日志汇总

日志汇总是将经过日志预处理的日志从系统的各个节点上汇总起来。此阶段利用Flume将日志数据汇总，并以文本文件方式提交到HDFS接口。Flume是1个分布式的海量日志采集、聚合和传输系统，内置了对HDFS的支持，由1个或多个agent构成，每个agent包括来源source、传输通道channel、接收和发送的sink。通过配置agent的各个属性，将各节点的日志汇总到HDFS中。具体配置为：

```
#设置agent，指定各个属性
agent.sources=spooldirSource
agent.channels=memoryChannel
```

agent.sinks=hdfsSink #设置监视日志的保存目录/home/searchlogs/,这样当目录下的文件有变化时,就会自动发送到HDFS中

agent.sources.spooldirSource.type=spooldir

agent.sources.spooldirSource.spoolDir=/home/searchlogs/

agent.sources.spooldirSource.channels=memoryChannel

#设置sinks,指定HDFS的接收url地址和存储地址

agent.sinks.hdfsSink.type=hdfs

agent.sinks.hdfsSink.hdfs.path=hdfs://masternode:9000/flume/

agent.sinks.hdfsSink.channel=memoryChannel

#设置channel,指定以内存方式进行传输

agent.channels.memoryChannel.type=memory

agent.channels.memoryChannel.capacity=100

2.3 日志采集和存储

采用HDFS存储海量的日志。HDFS作为Hadoop框架的一部分,提供高吞吐量的数据访问和海量文件的存储,并对外提供便捷的调用接口。在日志汇总阶段,应用Flume调用HDFS接口将日志文件进行上传汇总。在日志挖掘中,也通过HDFS接口读写日志文件。

3 日志挖掘和聚类分析

本模块基于内容推荐技术,利用MapReduce分布式计算模型。采用网页的标签和导航建立整个网站网页的向量空间模型,利用用户打开的搜索结果网页的标签和导航建立起用户兴趣模型;根据用户兴趣模型使用K-means聚类算法,并结合MapReduce,将有相似兴趣的用户划分到1个兴趣组中,使系统只需面对有限个数的兴趣组,而不是面对每个用户来提供个性化服务。

3.1 网页的向量空间模型和用户兴趣模型

3.1.1 网页的向量空间模型

信息检索模型主要有布尔模型、概率模型、向量空间模型3种。布尔模型是基于集合论和布尔代数的一种简单检索模型,实现简单、计算速度快,但其特征项没有考虑权重,且无法按照相关性进行量化计算和排序。概率模型是以数学理论中的概率论为原理的一种检索模型,有严格的数学理论为依据,但计算复杂度过高,不太适合大数据量计算。向量空间模型(vector space model,VSM)[7],是把对文本内容的处理转化为向量空间中的向量计算,通过计算向量之间的相似性来度量文档间的相似性,关键点是选取特征项和特征项权值、及相似度计算。向量空间模型能够量化计算相关性,十分便于大数据的并行计算。因此本系统选择向量空间模型。

1)选取特征项和特征项权值。在特征项的选取上,本系统使用网页的标签和导航。因为大部分政府网站按照国家的规定对每个网页都添加了标签和导航,标签描述了网页的主要内容,导航描述了各个网页间的层次关系,且其一级导航还用来协助确定K-means算法中的聚类个数和初始聚类中心。因此对于所有的网页,都可以用向量空间模型来表示,具体表示为:$D_i=(T_{i1},W_{i1};T_{i2},W_{i2};\cdots;T_{in},W_{in})$,其中$T_{ik}$为特征项,即此网页的标签;$W_{ik}$为特征项权值,其计算利用TF-IDF公式[8]:

$$W_{ik}(d)=\frac{tf_{ik}(d)\lg\left(\frac{N}{n_k}+0.01\right)}{\sqrt{\sum_{k=1}^{n}\left(tf_{ik}(d)\right)^2\times\lg^2\left(\frac{N}{n_k}+0.01\right)}} \quad (1)$$

其中,$tf_{ij}(d)$表示特征词T_{ij}在网页d中出现的频率,N为所有网页的个数,n_i表示网页中含有特征词T_{ij}的网页个数。

2)相似度计算。向量空间模型常用的相似性计算方法包括余弦距离、欧几里德距离等,由于计算聚类分析时更注重从方向上区分差异,而不是绝对数值的差异,因此本系统采用余弦距离公式:

$$\mathrm{Sim}(X,\ Y)=\cos(X,\ Y)=\frac{\sum_{k=1}^{n}\left(W_{xk}\times W_{yk}\right)}{\sqrt{\sum_{k=1}^{n}W_{xk}^2}\sqrt{\sum_{k=1}^{n}W_{yk}^2}} \quad (2)$$

式中,X、Y分别为2个页面的矢量,$\mathrm{Sim}(X,Y)$为X向量和Y向量之间的夹角余弦,W_{xk}为X页面的第K个特征项权值,W_{yk}为Y页面第K个特征项权值。

3.1.2 用户兴趣模型

通过类似方法,将用户打开过的所有搜索结果的网页标签和导航转化为空间中的向量,从而得到用户的兴趣模型。用户如果打开搜索结果网页,就表示用户对这些标签和导航所代表的内容感兴趣,依次建立用户的向量空间模型,即用户的兴趣模型,表示为$U_i=(T_{i1},W_{i1};T_{i2},W_{i2};\cdots;T_{im},W_{im})$,其中特征项$T_{ik}$为此用户打开的所有网页的标签和导航,特征项权值$W_{ik}$仍用TF-IDF公式计算,代表该用户对此标签和导航感兴趣的程度,用户之间兴趣的相似度计算仍用余弦距离公式。

3.1.3 MapReduce分布式计算模型

本系统采用MapReduce分布式计算模型,建立向量空间模型并进行K-means聚类计算[9]。MapReduce是Hadoop框架的一部分,通过把一个复杂的任务分解为Map映射和Reduce归约操作,简化输入、分割、任务调度、数据通信等技术细节,可高效地对海量数据进行分布式计算,很适合对日志数据进行快速处理,从而建立网页的向量空间模型和用户兴趣模型。

MapReduce计算模型将任务初始化为一个Job,分成5个阶段:1)Input阶段把输入的文件切分并读入;2)Map阶段调用用户定义的Map函数,分析读入的文件内容,构造出<key,value>键值对;3)Shuffle/Sort阶段通过复制、分组、排序过程,把相同key值的value集合到一起,组合成<key,list<value>>;4)Reduce阶段对输入的<key,list<value>>,执行用户自定义的Reduce函数,输出新的<key,value>;5)Output阶段把新的<key,value>结果写入输出目录。其中主要是实现Map和Reduce两个自定义函数的阶段,其他阶段由MapReduce计算模型自动实现。流程如图3所示。

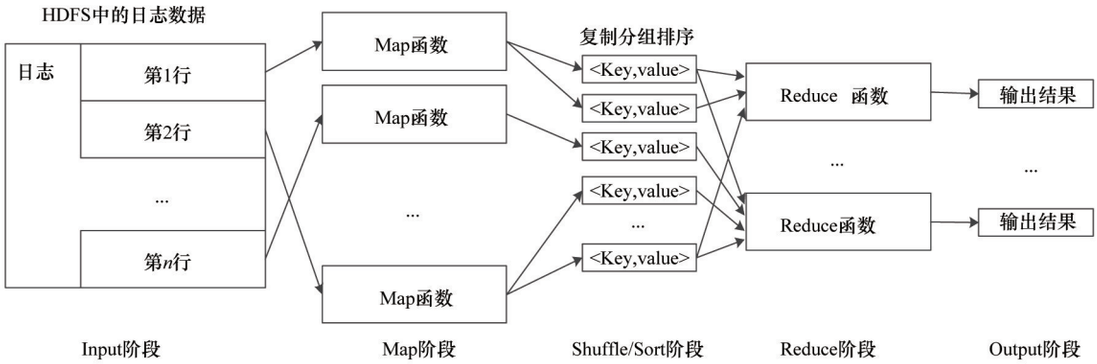

图3　MapReduce流程

Fig. 3　MapReduce flowchart

3.1.4　MapReduce计算网页向量空间模型和用户兴趣模型

计算网页向量空间模型和用户兴趣模型均需要利用MapReduce,本文着重介绍用户兴趣模型的建立。实现MapReduce计算,需要自定义Map和Reduce两个函数。

1）Map()函数按行读取日志数据处理模块保存在HDFS中的日志,把UserId和Label组成UserId#Label,来作为Key;将出现次数Count作为Value,每出现一次Count自增1。键值对<User#Label, Count>作为结果写入context,发送给Reduce函数。Map函为:

```
public static class MyMapClass extends Mapper<Object, Text, Text, IntWritable>{
//lineIndex是行号,value 是行内容,context用于输出Map结果
protected void map(LongWritable lineIndex, Text value, Context context){
final String[]splited = value.toString().split("#");
String userId=splited[0].split("=")[1];
String[]labels=splited[2].split(" =")[1].split(",");//需要分拆出每一个标签,分别计算
for(String label:labels) {
    Text record=new Text();//封装K值
    record.set(userId+"#"+label);//Key的格式为userId#label
    context.write(record, new IntWritable(1));//Value值为1
}}};
```

2）Reduce()函数根据User#Label进行归并,相同User#Label的Value值求和,从而计算出某用户打开的网页中此标签出现的总次数,次数越多权重越大,最后根据式(1)计算用户的兴趣模型$\boldsymbol{U}_i$,为方便存储,关联用户UserId,定义为键值对<UserId, $\boldsymbol{U}_i$>。

```
public static class MyReduceClass extends Reducer<Text, IntWritable,Text, IntWritable>{
private IntWritable result=new IntWritable();//保存本次Reduce的结果
//key为userId#labels;values是多个Map()计算值的集合
public void reduce(Text key, Iterable<IntWritable>values, Context context){
int tmp=0;
  for (IntWritable val: values){
    tmp=tmp+val.get();
  }
  result.set(tmp);//将多个Map的值求和
  context.write(key, result);//输出最后的汇总结果
}}
```

通过计算得到用户的兴趣模型后,采用同样的步骤,将url#label作为Key,就可得到网页向量空间模型<Url, $\boldsymbol{D}_i$>。

经过以上处理,将<UserId, $\boldsymbol{U}_i$>和<Url, $\boldsymbol{D}_i$>保存到HDFS中,用于下一步的聚类分析。

3.2　聚类分析

聚类分析是数据挖掘领域中常用的算法,是把相似的对象通过静态分类的方法分成不同的组别或者更多的子集,使同一个子集中的对象之间相似性更高。本模块基于上文建立的用户兴趣模型,使用聚类分析中的*K*-means算法,将兴趣相似的用户组成一个兴趣组。

3.2.1　*K*-means算法

聚类算法包括很多种,如划分聚类算法、层次聚类算法、基于模型的聚类算法等。以上聚类算法根据特性不同分别有不同的适用领域。在日志挖掘领域,对聚类算法的要求是,算法要有良好的伸缩性,能够处理*G*为数量级的Web日志数据,算法能减少人为选择的初始化参数,提高计算结果的客观性。根据上述要求,本模块选择广泛使用的基于划分的聚类算法*K*-means算法。该算法适合大数据量的并行计算,通过结合政府网站特点能够有效的选择初始化参数,可尽量避免人为选择的影响。该算法认为,簇是由距离靠近的对象组成的,对象间的距离越近相似度就越大。*K*-means算法把*N*个对象分为*K*个簇,簇内具有较高的相似度,算法的主要步骤为:

步骤 1: 从数据集合 N 中选取 K 个数据作为初始聚类中心：$C_i(A)$,$A=i$,（$i=1,2,\cdots,K$）；

步骤 2: 计算各数据对象与每个聚类中心的距离：$D(X_i, C_j(A))$,（$i=1,2,\cdots,N$; $j=1,2,\cdots,K$）；

步骤 3:将各个对象 i 配给距离最近（相似度最高）的簇，满足 $D(X_i, C_j(A))=\min\{D(X_i, C_j(A)), j=1,2,\cdots,K\}$,将其归类到聚类中心 $C_j(A)$；

步骤 4: 更新各个簇的聚类中心和平均值 $C_j(A+1)$；

步骤 5: 判断 $C_j(A)$是否与 $C_j(A+1)$的距离是否小于一定的阈值，如果是则到步骤 6结束；如果否，则返回步骤 2开始新一轮迭代；

步骤 6: 输出聚类结果。

根据上述步骤可以看出，K-means 迭代计算每个样本与质心之间距离时，迭代过程为$(N\times K)$次，N 是总样本数，K 是质心数，由于每个样本与质心之间距离的计算是相互独立的，因此算法具有良好的伸缩性，能充分使用并行计算来处理海量数据。但 K-means 算法存在聚类个数 K 难以确定、初始聚类中心难以选择 2 个明显缺点，且都与初始值的选择有关。对这 2 点如果选择不当，就会影响结果的准确性。

由于政府网站具有多级目录层次结构，且导航是按目录层次设定的，因此本系统选择政府网站的一级导航作为聚类中心点，一级导航个数作为聚类个数并以此作为 K-means 计算的初始值。

3.2.2 利用 MapReduce 计算 K-means 聚类

根据计算出的用户兴趣模型，再次利用 MapReduce 进行 K-means 聚类计算[10~13]，从而将有相似兴趣的用户划分到一起组成一个兴趣组。方法是将 K-means 算法分解，其中 Map 函数负责将每个<UserId, U_i>放入离它最近的簇中，Reduce 函数实现质心的更新，得到新的聚类中心，具体流程为：

1）Map 函数将上述计算结果<UserId, U_i>作为输入，利用余弦距离公式计算数据点到各个聚类中心的距离，选择距离最近的聚类中心作为所属的簇。将此 U_i 属于的第 n 个质心 C_n 作为 Key，将此用户的 U_i 作为 Value，组成的输出键值对为 $<C_n, U_i>$，执行流程如图 4 所示。

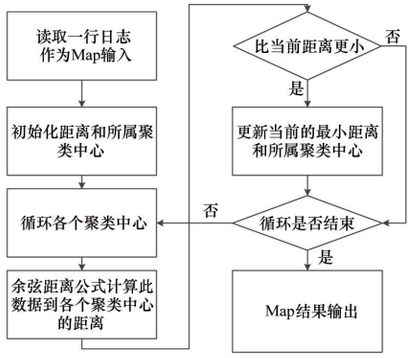

图 4 Map 函数的执行流程

Fig. 4 Map function execution flow chart

2）Reduce 函数将$<C_n$, List$<U_i>>$作为其输入，对相同簇中的所有的 U_i 进行求和计算，得到新的聚类中心，即新质心$<n, C_n>$。执行流程如图 5 所示。

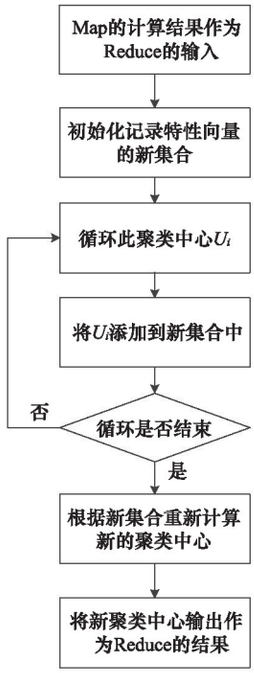

图 5 Reduce 函数的执行流程

Fig. 5 Reduce function execution flow chart

3）判断该聚类是否已收敛。每次迭代计算聚类中心后，计算上一轮 MapReduce 迭代过程后聚类中心与本次聚类中心的之间的距离，若大于给定阈值，则用本轮的聚类中心作为新的聚类中心，启动新一轮的 MapReduce 计算，反之则迭代结束，用户的兴趣模型建立完毕。

通过上述聚类计算，将用户按兴趣分到不同的兴趣组中，这样本系统就可以针对兴趣组来提供个性化搜索和个性化推送。

4 个性化搜索和推送

当确定用户的兴趣组后，判断用户对某网页是否感兴趣，只需根据网页的标签和导航，计算其到此用户所在聚类中心的距离。本系统以聚类的平均半径——簇内所有数据点到中心距离的平均值作为阈值，当小于此阈值，即认定用户对此搜索结果网页感兴趣，据此调整搜索结果的排序，当政府网站更新时将用户感兴趣的新网页推送给用户，从而实现个性化搜索和推送。

4.1 个性化搜索

用户进行搜索时，本系统首先获取用户所在的兴趣组，得到此兴趣组的质心 C_n 和平均半径 r_n，然后根据结果中各个网页到 C_n 的距离是否小于 r_n，判断是否是用户感兴趣的搜索结果网页，从而调整原有的结果排序，将用户感兴趣的网页排在搜索结果前列，流程如图 6 所示。

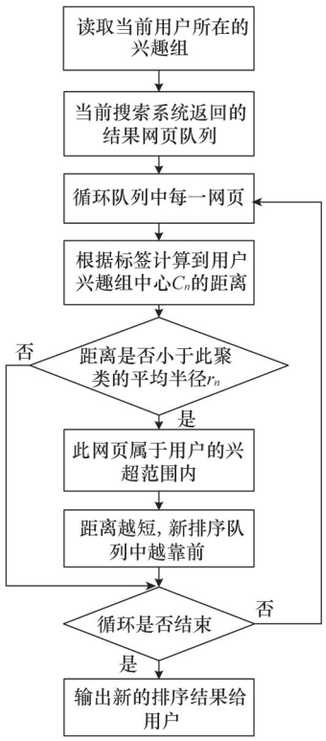

图6　个性化搜索排序调整

Fig. 6　Personalized search sort adjust map

4.2　个性化推送

当政府网站的页面更新时，本系统会主动将用户感兴趣的网页推送给用户，判断用户感兴趣网页的计算方法与个性化搜索相同。由于微博、微信和IOS有各自的接口来实现信息的推送，本文不再展开。本系统主要实现了基于Android系统的App推送功能。

4.2.1　Android的推送原理

Android有2种方法[13]实现推送：第1种是利用谷歌的Android Cloud to Device Messaging（C2DM）服务来推送消息；第2种是自建服务器，采取XMPP（可扩展消息处理现场协议）协议推送。XMPP是基于可扩展标记语言（XML）的协议，基于长连接方式实现向联网用户的App推送消息，为更好的控制消息的发送和接收，本系统采用此方式来实现推送服务。

本系统采用基于XMPP协议的开源框架androidpn来实现消息推送服务，它包含客户端和服务器端，简化了App客户端与服务器之间复杂的交互过程，实现了消息的推送。

4.2.2　推送服务器端的实现

服务器端首先要设置与App保持长连接的端口，以及推送的消息标题、内容和网页地址；其次，消息推送分为广播推送、指定用户推送2种，代码为：

```
String title="新政策发布实施";//新消息标题
String message="从下月初，**政策开始正式发布实施….";//新消息内容
String uri="http://www.***.gov.cn/zhengcefabu.html";//网页地址
String username="user1";//当指定用户推送时，接受新消息的用户名
NotificationManager notify=new NotificationManager();
notify.sendBroadcast(apiKey, title, message, uri);//广播推送，推送给所有用户
notify.sendNotifcationToUser(apiKey, username, title, message, uri);//指定用户推送
```

4.2.3　推送客户端的实现

客户端包含消息的收发、解析及长连接的发起、重连等功能，首先在配置文件中，设置好服务器的域名地址和端口：

```
xmppHost=xmpp.***.gov.cn  //推送服务器的域名
xmppPort=5222 //推送服务器的端口
```

在Android的配置文件AndroidManifest.xml中，设置好通知服务：

```
<service android:enabled="true"
    android:name="org.androidpn.client.NotificationService"
    android:label="NotificationService">
    <intent-filter>
        <action android:name="org.androidpn.client.NotificationService"/>
    </intent-filter>
</service>
```

其次，在App程序中通过NotificationReceiver接收发送来的推送消息，显示到用户手机界面，提醒用户有新的消息，关键程序代码为：

```
public final class NotificationReceiver extends BroadcastReceiver {
    public void onReceive(Context context, Intent intent) {
    if(Constants.ACTION_SHOW_NOTIFICATION.equals(intent.getAction())){
    String title=intent.getStringExtra(Constants.TITLE);//消息标题
    String message=intent.getStringExtra(Constants.MESSAGE);//消息内容
      String uri=intent.getStringExtra(Constants. URI);//消息网页地址
    }}
```

通过上述步骤，就能方便的将新网页的内容迅速推送到用户的App上。

5　结论

设计开发了政府网站移动搜索的日志挖掘和个性化定制系统，为用户供个性化的搜索和推送服务。系统已部署到某网站进行试运行，硬件由6台试验机组成，cpu分别为8核到24核的志强处理器，内存为8G到32G；应用软件为包含HDFS、MapReduce的Hadoop 2.2.0和Flume 1.4.0。试运行期

间,系统实时将日志数据自动采集汇总并存储,通过每天凌晨的定时任务,系统分析计算前1天的日志,采用迭代的方式更新用户兴趣组和网页向量空间模型,从而对外提供个性化搜索和推送服务。经对比试验,一个经常搜索清洁能源相关内容的手机用户,在个性化搜索结果中,网页标签中带太阳能、风能等与可再生能源有关的网页,其排序比原有的搜索结果都提前了,用户输入更少的查询词无需翻页就能看到感兴趣的搜索结果;并且有关新能源的新网页,都会在第2天主动推送到用户手机上,增加了与用户的互动。

总之,通过本系统提高了移动搜索的效果,增强了与用户的互动。本系统下一步将进一步改进功能,例如根据网页特点优化聚类算法,结合协同过滤,使有相似兴趣的用户能够互相推荐网页,并完善个性化定制功能。

参考文献(References)

[1] 中国互联网络信息中心. 第34次中国互联网络发展状况统计报告[EB/OL]. 2014-07-21[2014-08-20]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201407/P020140721507223212132.pdf.
China Internet Network Information Center. The 34th statistical report on internet development in China[EB/OL]. 2014-07-21[2014-08-20]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201407/P020140721507223212132.pdf.

[2] 王继民, 李雷明子, 郑玉凤. 基于日志挖掘的移动搜索用户行为研究综述[J]. 情报理论与实践, 2014, 37(3): 134-139.
Wang Jimin, Li Leimingzi, Zheng Yufeng. Review on mobile users search behavior based on Web log mining[J]. Information Studies: Theory & Application, 2014, 37(3): 134-139.

[3] 万飞, 赵溪, 梁循, 等. 基于移动互联网日志的搜索引擎用户行为研究[J]. 中文信息学报, 2014, 28(2): 144-150.
Wan Fei, Zhao Xi, Liang Xun, et al. Research on search engine mobile Internet user behavior based on log[J]. Journal of Chinese Information Processing, 2014, 28(2): 144-150.

[4] 赵龙. 基于hadoop的海量搜索日志分析平台的设计和实现[D]. 大连: 大连理工大学, 2013.
Zhao Long. The design and implementation of massive search logs analysis platform based on hadoop[D]. Dalian: Dalian University of Technology, 2013.

[5] 周婷婷. 基于海量查询日志的数据挖掘及用户行为分析[D]. 北京: 北京邮电大学, 2012.
Zhou Tingting. Data mining and user behavior analysis based on the massive query log[D]. Beijing: Beijing University of Posts and Telecommunications, 2012.

[6] 王振宇, 郭力. 基于Hadoop的搜索引擎用户行为分析[J]. 计算机工程与科学, 2011, 33(4): 115-120.
Wang Zhenyu, Guo Li. Search engine user behavior analysis based on Hadoop[J]. Computer Engineering & Science, 2011, 33(4): 115-120.

[7] 胡晓, 王理, 潘守慧. 基于改进VSM的Web文本分类方法[J]. 情报杂志, 2010, 29(5): 144-147.
Hu Xiao, Wang Li, Pan Shouhui. Web text classification method based on improved VSM[J]. Journal of Intelligence, 2010, 29(5): 144-147.

[8] 周炎涛, 唐剑波, 王家琴. 基于信息熵的改进TFIDF特征选择算法[J]. 计算机工程与应用, 2007, 43(35): 156-171.
Zhou Yantao, Tang Jianbo, Wang Jiaqin. Improved TFIDF feature selection algorithm based on information entropy[J]. Computer Engineering and Applications, 2007, 43(35): 156-171.

[9] 李杉, 刘莉莉. 基于MapReduce的Web日志挖掘[J]. 计算机工程与应用, 2012, 48(22): 95-98.
Li Shan, Liu Lili. MapReduce log mining based on Web[J]. Computer Engineering and Applications, 2012, 48(22): 95-98.

[10] Amresh K, Kiran M, Prathap B R. Verification and validation of mapreduce program model for parallel *K*-means algorithm on hadoop cluster [C]// 2013 Fourth International Conference on Computing, Communications and Networking Technologies. Tiruchengode, India: IEEE, 2013: 274-282.

[11] 江小平, 李成华, 向文, 等. *K*-means聚类算法的MapReduce并行化实现[J]. 华中科技大学学报: 自然科学版, 2011, 39(6): 120-124.
Jiang Xiaoping, Li Chenghua, Xiang Wen, et al. Parallel implementation of *K*-means clustering algorithm MapReduce[J]. Journal of Huazhong University of Science and Technology: Natural Science Edition, 2011, 39(6): 120-124.

[12] 周婷, 张君瑛, 罗成. 基于Hadoop的*K*-means聚类算法的实现[J]. 计算机工程与发展, 2013, 23(4): 18-21.
Zhou Ting, Zhang Junying, Luo Cheng. Realization of *K*-means clustering algorithm based on Hadoop[J]. Computer Technology and Development, 2013, 23(4): 18-21.

[13] 冀素琴, 石洪波. 基于MapReduce的*K*-means聚类集成[J]. 计算机工程, 2013, 39(9): 84-87.
Yi Suqin, Shi Hongbo. Clustering of *K*-means integration based on MapReduce[J]. Computer Engineering, 2013, 39(9): 84-87.

[14] 倪红军. 基于Android平台的消息推送研究与实现[J]. 实验室研究与探索, 2014, 33(5): 96-100.
Ni Hongjun. Research and implementation of push messages based on Android platform[J]. Research and Exploration in Laboratory, 2014, 38(5): 96-100.

(责任编辑 陈广仁)