基于双向图的关键词生成算法

杨艳[1,2]，朱敬华[1]，金虎[1]，李巍[1]

（1. 黑龙江大学 计算机科学技术学院，黑龙江 哈尔滨 150080；
2. 黑龙江大学 数据库与并行计算重点实验室，黑龙江 哈尔滨 150080）

摘 要：为了提高搜索引擎生成关键词的效果，提出基于双向图的关键词生成（BGKG，bipartite graph based keywords generation）算法。该算法基于搜索引擎日志生成关键词，利用日志中的关键词和点击网址建立一个双向图，考虑了搜索引擎返回网址的顺序和用户点击网址的顺序。在真实搜索引擎日志上实现了BGKG算法。大量实验的结果表明，BGKG算法生成的关键词能够代表企业客户的需求，在运行时间上也有明显优势。

关键词：搜索引擎；关键词生成；搜索引擎广告；日志；双向图

中图分类号：TP311 **文献标识码**：B **文章编号**：1000-436X(2011)9A-0146-07

Keywords generation algorithm based on bipartite graph

YANG Yan[1,2], ZHU Jing-hua[1], JIN Hu[1], LI Wei[1]

(1. Department of Computer Science and Technology, Heilongjiang University, Harbin 150080, China;
2. Key laboratory of Database and Parallel Computing, Heilongjiang University, Harbin 150080, China)

Abstract: To improve the efficiency of keywords generation, a bipartite graph based keywords generation (BGKG) algorithm was proposed. It generated keywords based on search engine logs and built a bipartite graph between query terms and the clicked URLs. It took into account the rank of the URLs in result pages and the order of users clicking. Experiments were done with real query logs. The results show that keywords generated by BGKG can satisfy the needs of enterprise clients and BGKG is more efficient than other keyword generation algorithms.

Key words: search engine; keyword generation; search engine advertising; log; bipartite graph

1 引言

随着网络和互联网的飞速发展，搜索引擎已经成为人们获取网络信息的主要途径。搜索引擎的盈利离不开广告[1]，关键词生成是搜索引擎广告的关键。关键词生成即生成满足特定企业客户需求的关键词集合，对搜索引擎服务商和企业客户都具有重要意义。

目前对关键词的研究主要集中在查询关键词扩展和关键词生成2个方面。在查询关键词扩展方面已经有大量的研究成果，包括基于向量空间的查询词聚类方法[2~4]、基于会话的查询关键词距离计算方法[5]、基于时间的语义相似性研究方法[6]、基于图的语义相似性研究方法[7]、基于关联规则的查询扩展方法[8]等。在关键词生成方面也出现了一些研究成果，归纳为4类。①基于元标记的方法。在

收稿日期：2011-07-05

基金项目：国家自然科学基金资助项目（60973081）；黑龙江省自然科学基金资助项目（F201011）；黑龙江省教育厅科学技术研究面上基金资助项目（11551352）

Foundation Items: The National Natural Science Foundation of China (60973081); The Natural Science Foundation of Heilongjiang Province (F201011); The Scientific Research Foundation of Heilongjiang Provincial Education Department (11551352)

网页代码中，元标记用 keywords 属性来概括描述该网页的主要含义。搜索引擎爬行网页时，可获得该属性，从而将所得结果作为关键词。②基于重复查询的方法。将企业客户所给的关键词与搜索引擎数据库中的查询词进行比对并扩展，将结果显示给企业客户。③基于相似性查询的方法。搜索引擎根据客户所给查询词进行查询，然后爬行搜索结果靠前的网页，从网页中找出与客户所给查询相近的关键词[9,10]。④基于搜索引擎日志的方法。根据搜索引擎日志生成与客户相关的关键词。上述方法仅能生成包含输入词词干的关键词，没有与输入词相关而不包含词干的关键词，即无法挖掘潜在的关键词。文献[11]提出基于搜索引擎日志的马尔科夫链关键词生成算法(ARW 算法)，此方法能生成潜在关键词，但没有考虑搜索引擎日志中网址的排名和用户点击的顺序。另外，ARW 算法在每次迭代过程中需要访问全部搜索引擎日志，时间开销较大。

本文在搜索引擎日志的基础上，提出基于双向图的关键词生成（BGKG，bipartite graph based keywords generation）算法。BGKG 算法在迭代过程中不需要访问全部搜索引擎日志，同时考虑了结果网址的排序和搜索引擎用户点击网址的顺序。实验表明算法生成的关键词能够满足企业客户需求，且在生成关键词集合基本相同的情况下，运行时间明显少于 ARW 算法。

2　问题描述

企业客户为了做广告，需将网址集合或关键词集合交给搜索引擎服务商，搜索引擎服务商使用搜索引擎日志和企业客户所给的网址或关键词，生成相关的关键词集合。

把企业客户提交的关键词集合或网址集合称为种子集合，定义如下：

定义 1(种子集合)　BGKG 算法的输入，表示企业客户所给的网址集合（企业网址或和企业相关的网址）或关键词集合，可以是单个或多个网址或关键词，用 SS 表示。

搜索引擎服务商在关键词生成过程中使用的搜索引擎日志描述如下：

定义 2(搜索引擎日志)　搜索引擎日志表示为 SEL<T,U,Q,L,R,C>，其中，U 为用户，L 为网址，T 为用户 U 点击网址 L 的时间，Q 为用户输入的查询关键词，R 为网址 L 在返回结果中的排名，C 为用户点击网址 L 的顺序号。

BGKG 算法使用的日志由搜索引擎日志统计得到，称为实验日志。实验日志不考虑搜索引擎日志中的用户因素，定义如下：

定义 3(实验日志)　实验日志的表示形式为 EL<Q,L,F,S,C>，其中，Q 为查询关键词，L 为查询关键词 Q 时点击的一个网址，F 为查询关键词 Q 时点击网址 L 的总次数，S 为网址 L 在返回结果中的平均排名，C 为用户点击网址 L 的平均顺序。

3　相关度定义

搜索引擎日志记录了用户输入的查询关键词和点击的相应网址。可以认为查询关键词是对点击网址的一个概括描述。对于一个查询关键词，用户可能点击返回结果中的多个网址；对于同一个网址，用户可能通过不同的查询关键词得到。这样一个关键词可描述多个不同网址，同样一个网址可能由不同关键词描述。每个关键词可能从不同方面对网址进行概括描述，应有不同权重。本文在考虑返回网址排名和用户点击顺序的基础上，提出新的关键词权重定义方法以及关键词与种子集合相关度的定义方法。

定义 4(网址的点击权重)　用户查询 q_i 时，点击的网址 u_j 在所有点击网址中的比重 $weight(q_i,u_j)=freq(q_i,u_j)/sum(q_i)$。其中，$freq(q_i,u_j)$表示搜索引擎用户在查询关键词 q_i 时点击网址 u_j 的次数，$sum(q_i)$表示搜索引擎用户在查询关键词 q_i 时点击网址的总数。

定义 5(搜索引擎认为关键词描述网址的权重)　搜索引擎认为关键词 q_i 描述网址 u_j 的权重 $SWeight(q_i,u_j)=1/S$。其中，S 表示在查询关键词 q_i 时，网址 u_j 在返回结果中的平均排名。平均排名越大则搜索引擎认为网址 u_j 与关键词 q_i 越不相关。

定义 6(搜索引擎用户认为关键词描述网址的权重)　搜索引擎用户认为关键词 q_i 描述网址 u_j 的权重 $CWeight(q_i,u_j)=(1-\theta_2)/C+\theta_2 Weight(q_i,u_j)$。其中，C 表示在查询 q_i 时，用户点击网址 u_j 的平均顺序，$Weight(q_i,u_j)$为用户查询 q_i 时，点击的网址 u_j 在所有点击网址中的比重，θ_2 为权重系数。

定义 7(关键词描述网址的权重)　关键词 q_i 描述网址 u_j 的权重 $SCWeight(q_i,u_j)=(1-\theta_1)SWeight(q_i,u_j)+\theta_1 CWeight(q_i,u_j)$。其中，$SWeight(q_i,u_j)$表示搜索引擎认为 q_i 描述 u_j 的权重，$CWeight(q_i,u_j)$表示搜索

引擎用户认为 q_i 描述 u_j 的权重，θ_1 为权重系数。

定义 8（关键词与种子集合的相关度） 关键词 q_i 与种子集合 SS 的相关度

$$QW(q_i)=\begin{cases}1, & q_i \in SS\\ UW(u_j)\,SCW eight(q_i,u_j), & q_i \notin SS\end{cases}$$

其中，网址 u_j 为用户查询 q_i 时点击的网址，且是将 q_i 添加到关键词集合的网址，$SCW eight(q_i,u_j)$ 表示 q_i 描述网址 u_j 的权重，$UW(u_j)$ 为网址 u_j 与种子集合的相关度。

定义 9（网址与种子集合的相关度） 网址 u_j 与种子集合 SS 的相关度

$$UW(u_j)=\begin{cases}1, & u_j \in SS\\ \left\{\frac{1}{n}\sum QW(q_i)\,|\,q_i \in Uqueryset(u_j)\right\}, & u_j \notin SS\end{cases}$$

其中，$Uqueryset(u_j)$ 表示点击网址 u_j 的查询关键词集合且该集合中的每个关键词与种子集合的相关度已经计算得出，n 为 $Uqueryset(u_j)$ 集合的大小。

4 关键词生成算法 BGKG

实验日志中每条记录都包含了查询关键词和搜索引擎用户点击的网址，它们之间形成一条边，关键词和网址之间就建立起相互连接的双向图。BGKG 算法利用该双向图，从种子集合出发，得到相应的关键词。当种子集合为网址时，从双向图中的网址出发，得到与网址连接的关键词，再从得到的关键词出发得到与关键词连接的网址，一直递归循环下去，直到找到所有满足条件的关键词。在迭代过程中，BGKG 算法利用索引技术，不需要每次扫描全部数据库，使算法效率得到提高。另外，当关键词集合不再增长时，算法结束，而不需要扫描整个数据库。

为了滤掉与种子集合相关度较低的关键词，算法设置了查询词阈值 λ。当得到的关键词与种子集合的相关度小于 λ 时，说明该关键词是经过很多步迭代得到的，相关度低，不予推荐。

算法 1 当种子集合为网址集合时的 BGKG 算法

输入：种子集合 SS；实验日志 EL；参数 θ_1、θ_2 和 λ。

输出：生成的关键词集合 VquerySet。

```
1) 生成的关键词集合 VquerySet=F ;
2) 生成的网址集合 URLSet=F ;
3) for all  u∈种子集合 SS  do
4)     UW (u)=1;
5)     u.visited=false;
6)     URLSet.add(u);
7) end for
8) addSet(VquerySet,URLSet);
9) return VquerySet.
```

算法 1 分 2 步：①初始化种子集合（3~7 行）：从企业客户所给的种子集合 SS 开始，将其中的每个网址 u 与种子集合 SS 的相关度 UW (u)置为 1；将每个网址的访问值置为假；并将每个网址放入网址集合 URLSet；②addSet(VquerySet,URLSet)函数从 URLSet 出发得到关键词集合 VquerySet（第 8 行）。

算法 2 当种子集合为关键词集合时的 BGKG 算法

输入：种子集合 SS；实验日志 EL；参数 θ_1、θ_2 和 λ。

输出：生成的关键词集合 VquerySet。

```
1)  生成的关键词集合 VquerySet=F ;
2)  生成的网址集合 URLSet=F ;
3)  for all  query∈种子集合 SS  do
4)      QW (query)=1;
5)      query.visited=true;
6)      VquerySet.add(query);
7)  end for
8)  for all query∈VquerySet do
9)       find urls which connect with query;
10)   for all u∈urls do
11)       if  u  URLSet  then
12)       Compute UW (u);
13)           u.visited=false;
14)       URLSet.add(u);
15)       end if
16)   end for
17) end for
18) VquerySet=F ;
19) addSet(VquerySet,URLSet);
20) return VquerySet.
```

算法 2 分 3 步：①初始化种子集合(3~7 行)：从企业客户所给种子集合 SS 开始，将 SS 内每个关键词与种子集合的相关度置 1，将每个关键词的访问值置为真并将其加入到关键词集合 VquerySet；②计算双向图中与种子集合内关键词相连的每一个网址与种子集合的相关度(8~17 行)；③把 VquerySet

清空，即从生成的关键词集合中把已在用户给定种子集合中的关键词去除；调用 addSet(VquerySet, URLSet)得到从 URLSet 出发的关键词集合 VquerySet(18~19 行)。

函数 addSet(VquerySet,URLSet)以 URLSet 为输入网址集合，生成关键词集合 VquerySet，包括以下 3 步。①从网址集合中未访问过的网址出发找出符合条件的关键词：从网址集合开始，找出每个未访问过的网址所连接的关键词集合。对于其中每个关键词 q_i，若 q_i 不属于 VquerySet，则计算其与种子集合的相关度 QW(q_i)。如果 QW(q_i)大于关键词阈值 λ，则将 q_i 放入 VquerySet。网址连接的关键词全部处理后，将网址的访问值设置为真(2~15 行)。②由关键词集合中未访问过的关键词出发找到相连接的网址：从关键词集合开始，找出未访问过的关键词。对于每个未访问过的关键词，从实验日志中找出与其连接的所有网址。对于相连的每个网址 u，如果 u 不属于网址集合 URLSet，则计算其与种子集合的相关度 UW(u)，并将其访问值置为假，然后将 u 加入到网址集合 URLSet 中。与该关键词相连的网址全部处理后，将该关键词的访问值设置为真(16~28 行)。③判断关键词集合 VquerySet 的大小有无变化，如果有变化，则重新运行①和② 2 个步骤，否则算法停止。算法停止有 2 个方面原因：一是新生成关键词的权重小于所给的关键词阈值；二是没有新的关键词生成(29~30 行)。

```
addSet(VquerySet,URLSet)
1) size=VquerySet.size();
2) for all u∈URLSet do
3)    if u.visited = = false then
4)       for all query which connect with u do
5)          if query VquerySet then
6)             compute QW (query)
7)             if QW (query)>λ then
8)                query.visited=false;
9)                VquerySet.add(query);
10)               end if
11)          end if
12)      end for
13)      u.visited=true;
14)   end if
15) end for
16) for all query∈VquerySet do
17)   if query.visited = = false then
18)      find urls which connect with query
19)      for all u∈urls do
20)         if u URLSet then
21)            Compute UW (u);
22)            u.visited=false;
23)            URLSet.add(u);
24)         end if
25)      end for
26)      query.visited=true;
27)   end if
28) end for
29) if(size VquerySet.size())then goto 1;
30) end if
```

5　实验

实验利用 2007 年 3 月的搜狗搜索引擎日志进行。先将文本搜索引擎日志导入数据库，共 44 410 900 条日志记录，其中有 4 579 805 个不同的关键词和 14 976 375 个不同的网址。再按关键词和网址分组。对于网址出现次数小于等于 1 的组，认为是用户的偶然点击，不能代表用户对该网址感兴趣，予以删除，得到实验日志的数据库形式，共 3 462 970 条实验日志记录。最后分别对关键词和网址建立索引。实验在 Pentium(R) 4 3.00GHz 处理器的微机上运行，内存为 512MB，操作系统为 Windows XP，数据库为 ORACLE 10G。实验包括如下 3 方面。

1) 不同参数对算法效果的影响

分析参数 λ、θ_1、θ_2 对算法的影响。Q_i 表示选择第 i 组参数时生成的关键词集合，$|Q_i|$为 Q_i 中关键词的数量，Q_i^k 表示 Q_i 的前 k 个关键词集合。下面的实验中 $k=|Q_i|$。

由表 1 的 $|Q_i|$列可知，随着相关度阈值 λ 的减小，所得关键词集合不断增大。考察 $|Q_i|/|Q_{i+1}|$列和 $|Q_i\cap Q_{i+1}|/|Q_{i+1}|$列，由 i=1 和 i=2 2 行($\lambda$=0.5 和 λ=0.3 行)取值对应相等说明 Q_1 Q_2 且 Q_2 Q_3，而 i=3 和 i=4 2 行(λ=0.1 和 λ=0.08 行)对应的取值不等说明 Q_3 Q_4 且 Q_4 Q_5。这说明 λ 取值较高时，生成的关键词比较稳定，只是在原有关键词的基础上扩充新的关键词；λ 取值较低时，生成的关键词集合会发生较大变化。由 $|Q_i^k\cap Q_{i+1}^k|/|Q_i|$列可知，当 λ=0.5 时，Q_1 和 Q_2 集合的前 $k=|Q_1|$项完全相同，但随着 λ 递减，生成的关键词集合 Q_i 和 Q_{i+1} 的前 $k=|Q_i|$项不完全相

同。可见，随着λ递减，生成的关键词顺序有相应的变化，说明生成的关键词集合里的关键词的权重有所不同。

表1 $\theta_1=\theta_2=0.5$ 时 λ 值对实验结果的影响

i	λ	$\|Q_i\|$	$\|Q_i\|/\|Q_{i+1}\|$	$\|Q_i\cap Q_{i+1}\|/\|Q_{i+1}\|$	$\|Q_i^k\cap Q_{i+1}^k\|/\|Q_i\|$
1	0.5	12	25.53%	25.53%	100%
2	0.3	47	1.06%	1.06%	95.74%
3	0.1	4 415	38.73%	35.14%	82.24%
4	0.08	11 398	47.22%	33.43%	69.39%
5	0.06	24 136	—	—	—

由表2的$|Q_i|$列可知，随着θ_1递减，所得关键词集合不断增大。分析$|Q_i|/|Q_{i+1}|$列和$|Q_i\cap Q_{i+1}|/|Q_{i+1}|$列，当i=1(即$\theta_1=1$)时，二者取值差别较大，考察i=3和i=4 2行($\theta_1=0.5$和$\theta_1=0.3$)，$|Q_i\cap Q_{i+1}|/|Q_{i+1}|$与$|Q_i|/|Q_{i+1}|$接近相等，说明生成的关键词比较稳定。考察$|Q_i^k\cap Q_{i+1}^k|/|Q_i|$列，$|Q_3^k\cap Q_4^k|/|Q_3|$和$|Q_4^k\cap Q_5^k|/|Q_4|$取值接近100%，说明生成的关键词顺序变化较小。

表2 $\lambda=0.1$,$\theta_2=0.5$ 时 θ_1 值对实验结果的影响

i	θ_1	$\|Q_i\|$	$\|Q_i\|/\|Q_{i+1}\|$	$\|Q_i\cap Q_{i+1}\|/\|Q_{i+1}\|$	$\|Q_i^k\cap Q_{i+1}^k\|/\|Q_i\|$
1	1.0	1 922	73.27%	64.13%	74.92%
2	0.7	2 623	59.41%	57.10%	81.32%
3	0.5	4 415	47.98%	46.99%	96.24%
4	0.3	9 202	44.94%	43.94%	97.34%
5	0.0	20 476	—	—	—

由表3的$|Q_i|$列可知，随着θ_2递减，所得关键词集合不断增大。分析$|Q_i|/|Q_{i+1}|$列和$|Q_i\cap Q_{i+1}|/|Q_{i+1}|$列，除$\theta_2=1$和$\theta_2=0$行，其他行二者的取值都几乎相等，说明生成的关键词比较稳定，而对于$\theta_2=1$行，这2列的取值相差1.12%，差别不大。考察$|Q_i^k\cap Q_{i+1}^k|/|Q_i|$列，$\theta_2=0.5$和$\theta_2=0.3$时$|Q_3^k\cap Q_4^k|/|Q_3|$和$|Q_4^k\cap Q_5^k|/|Q_4|$取值接近100%，说明生成的关键词顺序变化较小。

表3 $\lambda=0.1$,$\theta_1=0.5$ 时 θ_2 值对实验结果的影响

i	θ_2	$\|Q_i\|$	$\|Q_i\|/\|Q_{i+1}\|$	$\|Q_i\cap Q_{i+1}\|/\|Q_{i+1}\|$	$\|Q_i^k\cap Q_{i+1}^k\|/\|Q_i\|$
1	1.0	56	12.31%	11.43%	89.29%
2	0.7	455	10.31%	10.03%	70.11%
3	0.5	4 415	57.74%	57.72%	96.38%
4	0.3	7 647	48.31%	48.21%	95.74%
5	0.0	15 830	—	—	—

从表2和表3可知：当$\theta_1=0.5$或0.3且$\theta_2=0.5$或0.3时，所得结果比较理想，而$\theta_1=1$或$\theta_2=1$时结果并不理想。这说明，在生成关键词时仅考虑搜索引擎用户的因素或仅考虑搜索引擎的因素都是不合理的，本文算法同时考虑这2个因素是有必要的。

2) BGKG 算法的实验验证

关键词生成算法的评价采用了文献[11]的评价方法：相关和非直接相关。

相关：使用相关率$R(Q_K)=|Q_R|/|Q_K|$来表示生成的关键词集合与种子集合的相关度。其中，Q_K为关键词集合Q的前K个关键词，Q_R为Q_K中与种子集合相关的关键词集合。

非直接相关：非直接相关率反映了算法挖掘潜在关键词的能力。非直接相关率$N(Q_K)=|Q_K-Q(S)|/|Q_K|$。其中，Q_K表示生成的关键词集合Q的前K个关键词，Q(S)表示关键词集合Q_K中与种子集合直接相关的关键词集合，则任意不属于Q(S)的关键词与种子集合非直接相关。

关键词生成评价采用专家测试法，对关键词集合的前K个关键词进行测试，如表4所示。

表4 相关与非直接相关

K	$R(Q_K)$	$N(Q_K)$
10	100%	20%
50	98%	26%
100	92%	40%
500	86.6%	50.2%
1 000	75.2%	64.5%

由上表可知，随着生成的关键词集合数量的增加，生成关键词的相关度会递减，但非直接相关的关键词在增加。这说明算法可以生成不直接相关的关键词，即潜在关键词。并且随着生成的关键词数量的增加，非直接相关的关键词的数量也随之增加。

3) BGKG 算法和其他算法的比较

ARW 算法是目前基于搜索引擎日志的关键词生成最好的算法。BGKG 算法与ARW 算法分别从生成的关键词集合和算法运行时间2方面进行实验对比。实验中首先用BGKG 算法生成关键词集合，然后用ARW 算法生成相同数量的关键词集合进行对比。

BGKG 算法与ARW 算法采用相同的种子URL集合，分别生成关键词集合Q_1、Q_2和Q_3，然后对

相对应的关键词集合进行比较，其中，$|Q_1|=4\,415$、$|Q_2|=11\,398$、$|Q_3|=24\,136$。$|BGKG \cap ARW|/|BGKG|$表示 BGKG 算法与 ARW 算法生成的关键词集合的交集与 BGKG 算法生成的关键词个数的比值，用来验证生成相同数目的关键词时，有多少公共的关键词。

表 5 表明 BGKG 与 ARW 生成的关键词集合基本相同。通过领域专家对关键词的分析，BGKG 算法不仅能生成直接相关的关键词，还能发现新的语义相关的潜在关键词。

表 5　　公共关键词个数对比

关键词集合	$\|BGKG \cap ARW\|/\|BGKG\|$
Q_1	92.14%
Q_2	94.39%
Q_3	94.87%

算法运行时间是利用 BGKG 和 ARW 算法生成同样大小的关键词集合进行的比较。在 BGKG 算法中，对关键词和网址进行了分组聚集并建立索引，因此在算法运行过程中不需要扫描整个数据库，使算法效率显著提高。2 个算法的运行时间相差 2 个数量级，因此算法运行时间在不同的图中展示。

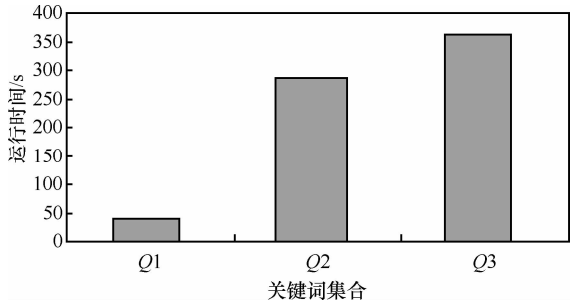


图 1　BGKG 算法运行时间

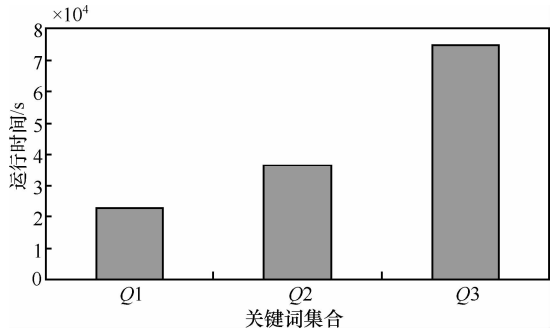


图 2　ARW 算法运行时间

图 1 和图 2 表明，在生成关键词个数相同的情况下，BGKG 运行时间比 ARW 有明显优势，且随着生成关键词数量的增加，效果更加明显。分析其原因，ARW 需多次扫描全部日志，而 BGKG 只需扫描部分日志。

6　结束语

提出基于搜索引擎日志的关键词生成算法 BGKG，考虑了用户点击网址的顺序和点击网址的排名。实验证明这 2 个因素对关键词生成有较大影响。算法在实现方面采用基于索引的双向图搜索技术。大量实验表明，BGKG 算法在运行效率和效果 2 方面都明显优于 ARW 算法，不仅生成了一些直接相关的关键词，还生成了潜在的关键词。

参考文献：

[1] 彭强.Google 的三种利器——技术、口碑、竞价广告[J]. 成功营销, 2003,(11):75-77.
PENG Q. Three kinds of edge tools of Google: techniques, public praise and advertisements[J]. Successful Marketing, 2003, (11):75-77.

[2] RICARDO B Y. Applications of Web Query Mining[M]. Heidelberg: Springer Berlin, 2005.

[3] RICARDO B Y, CARLOS H, MARCELO M. Query clustering for boosting web page ranking[A]. Proceedings of 2004 Atlantic Web Intelligence Conference[C]. Mexico: Springer, 2004.164-175.

[4] RICARDO B Y, CARLOS H, MARCELO M. Query recommendation using query logs in search engines[A]. International Workshop on Clustering Information over the Web[C]. Heraklion-Crete, Greece: Springer, 2004.588-596.

[5] ZHANG Z Y, NASRAOUI O. Mining search engine query logs for query recommendation[A]. Proceedings of the 15th World Wide Web Conference[C]. Edinburgh Scotland: ACM Press, 2006.1039-1040.

[6] ZHAO Q, HOI S, LIU T Y, et al. Time-dependent semantic similarity measure of queries using historical click-through data[A]. Proceedings of the 15th International Conference on World Wide Web[C]. Edinburgh Scotland: ACM Press, 2006.543-552.

[7] RICARDO B Y, ALESSANDRO T. Extracting semantic relations from query logs[A]. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York: ACM press, 2007.76-85.

[8] SHI X D, YANG C C. Mining related queries from web search engine query logs using an improved association rule[J]. Journal of the American Society for Information Science and Technology, 2007,

58(12):1871-1883.

[9] JOSHI A, MOTWANI R. Keyword generation for search engine advertising[A]. The 2006 IEEE International Conference on Data Mining[C]. Hong Kong: IEEE Computer Society, 2006.490-496.

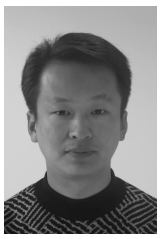
[10] YIH W, GOODMAN J, CARVALHO V. Finding advertising keywords on web pages[A]. Proceedings of the 15th International Conference on World Wide Web[C]. Edinburgh Scotland: ACM Press, 2006.213-222

[11] ARIEL F, PANAYIOANAYIOTIS T, KANNAN A, et al. Using the wisdom of the crowds for keyword generation[A]. Proceeding of the 17th International Conference on World Wide Web[C]. Beijing, China: ACM Press, 2008.61-70.

作者简介：

杨艳（1973-），女，黑龙江兰西人，博士，黑龙江大学教授、硕士生导师，主要研究方向为数据库和信息检索。

朱敬华（1976-），女，黑龙江齐齐哈尔人，博士，黑龙江大学副教授、硕士生导师，主要研究方向为数据库。



金虎（1972-），男，黑龙江依兰人，硕士，黑龙江大学副教授，主要研究方向为数据库和传感器网络。

李巍（1971-），男，山东寿光人，黑龙江大学工程师，主要研究方向为数据库。