Clustering Similar Query Sessions Toward Interactive Web Search

Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang

Department of Computer Science, National Taiwan University

Page 115 ~ 134

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

Clustering Similar Query Sessions
Toward Interactive Web Search

Chien-Kang Huang, Lee-Feng Chien*, Yen-Jen Oyang
Department of Computer Science, National Taiwan University, Taiwan.
*Institute of Information Science, Academic Sinica, Taiwan.
ckhuang@mars.csie.ntu.edu.tw, *lfchien@iis.sinica.edu.tw, yjoyang@csie.ntu.edu.tw

Abstract

A new effective log-based approach for interactive Web search is presented in this paper. The most important feature of the proposed approach is that the suggested terms corresponding to the user's query are extracted from similar query sessions, rather than from the contents of the retrieved documents. The experiment results demonstrate that this approach has a great potential in developing more effective web search utilities and may inspire more studies on advanced log mining mechanisms.

1. Introduction

Users' queries for Web search are usually short. For example, the average length of TREC topic description for conventional text retrieval is 15 tokens [11,12], while analyses of web search engine logs reveal that the average query length for Web search is about 2.3 tokens [6,9]. Short queries means that the information about the user's intention provided to the search engine is very limited. To deal with the short query problem, interactive search techniques [2,7] which attempt to identify the user's intentions and suggest more precise query terms are therefore commonly incorporated in Web search engine design.

To determine more relevant query terms for each given query, the conventional interactive search processes often rely on the key terms in the retrieved documents [2,7,10]. The key term set is extracted either statically from the documents during preprocessing or dynamically on-the-fly. Since the precision rates of the retrieved documents are usually not high enough, the extracted key terms are often found not relevant and not very helpful in practical Web search services.

In fact, extraction of relevant terms can be carried out by analyzing users' logs. In recent years, mining search engine logs has been obtaining more attention. Silverstein et al. [9] performed a second-order analysis on a log with a huge number of Web query terms. The results are then used to facilitate phrase recognition and query expansion [3].

In this paper, we propose a new approach based on log analysis for developing more effective interactive Web search engines. The most important feature of the proposed approach is that the suggested terms are extracted from similar query sessions, rather than from the contents of the retrieved documents. A query session is defined as a sequence of search requests issued by a user for a certain information need. The basis of the proposed approach is that two users with the same information need will issue common or related query terms. For example, in search for a subject regarding "search engine technology", a user may submit query terms such as "search engine", "Web search", "Google", "Web search and multimedia", while another user may submit "Web search", "Lycos". Therefore, if similar query sessions could be identified, query terms for the same information need can be extracted and applied to improve the effectiveness of search engines.

The remainder of the paper will be organized as follows. Section 2 is a brief introduction to the idea of interactive search based on similar query sessions. The method proposed for segmenting query sessions from proxy logs will be described in Section 3. Then, how query sessions are clustered is addressed in Section 4. Section 5 will present some experiment results and a conclusion is given in Section 6.

2. Interactive Search Based on Similar Query Sessions

Fig.1 is an abstract diagram showing our idea for interactive search. Before introducing the basic idea of the proposed approach, the concept of query session is presented and defined below:

Definition of Query Session:

Query session = (ID, R_1,…, R_m) where ID means the identifier of a user submitting a sequence of requests to a search engine in a certain period of time. Each request R_i = (t_i, q_i) means user ID sends a query term q to the search engine at time t

The proposed approach is assumed that the query space of users is formed by clusters of users' query sessions, and a set of query sessions grouped in the same clusters contain similar information needs. For each input query session with a sequence of *i* query terms, the interactive search process is then designed to retrieve the most similar cluster of query sessions from the query space, and then extract relevant terms in the cluster as suggested terms for next search. Once the *i+1*th query term is selected, it forms a new query session with *i+1* terms and the interactive process will perform again.

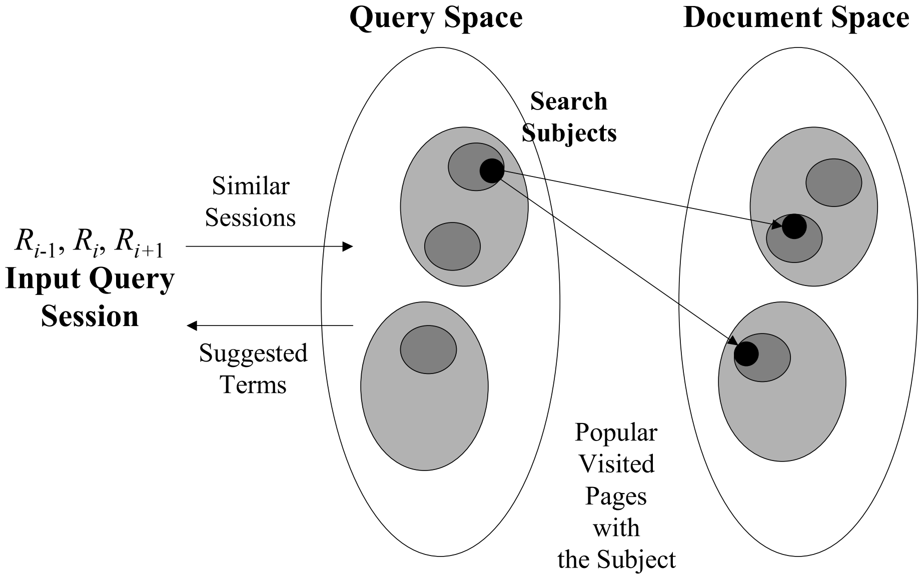

Fig.1 An abstract diagram showing our idea for interactive search.

Based on the above definition and idea, the problem to be dealt with is then formulated.

The Query Session Clustering Problem

For a set of query sessions from a query session log, the considering problem is to cluster these query sessions into different groups based on estimated similarity between query sessions. Each cluster can be defined as {S_i| f(S_i, S_j) > threshold}, in which f() is the similarity estimation function between query sessions.

Overview of the Proposed Approach

The proposed approach, as shown in Fig. 2, is composed of three processing modules: query session segmentation module, query session clustering module and relevant term extraction module. In the stage of query session segmentation, each query session will be segmented and extracted from a proxy log, according to the time gap between successive search requests. All of the extracted query sessions will form as a query session log. In the session clustering stage, the sessions with similar queries will be clustered and the cluster names extracted from composed high frequency terms. In the relevant term extraction stage, the relevance between the recorded query terms will be calculated and sets of relevant terms will be extracted for term suggestion applications in a search engine.

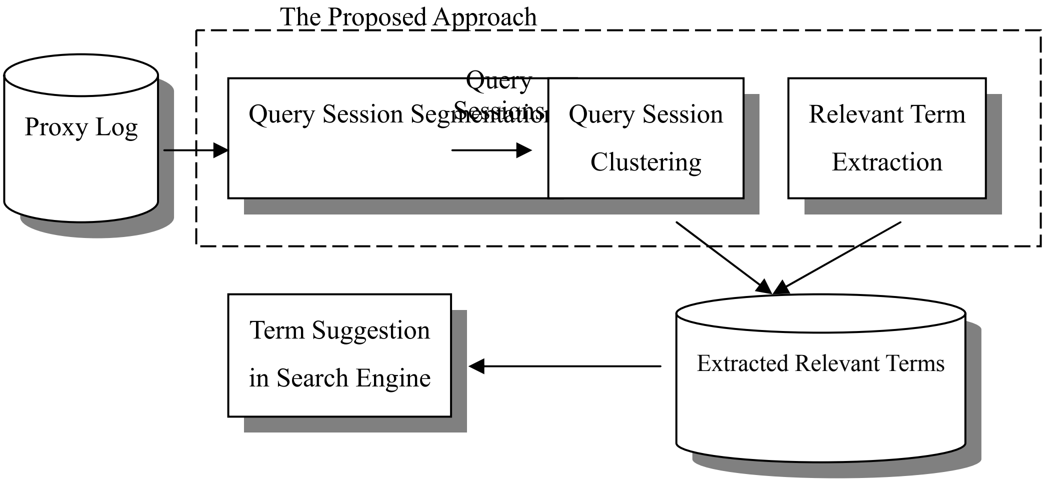

Fig.2 An overview of the proposed approach

3. Query Session Segmentation

A common proxy server might easily have thousands of clients accessing the web through it. Not only the general HTTP requests could pass through the proxy server, all of search HTTP requests are same. Compared with common search engine logs, a proxy server's log records more rigid information for users' information access and, more importantly, the recorded search requests are not limited to certain search engines.

However, a proxy log might record too much information and only some of them are useful in terms of search engine applications [13]. In our application it is sufficient to only use the following fields of logging information:

- A **timestamp** that indicates when a search request was submitted.
- A **client address** that indicates the IP address of the requesting instance.
- A **URL** string that contains the request content.

Since the experiments are just performing, the testing log is from NTU proxy servers and is still small. Some statistics of the testing proxy log are listed in Table 1.

Logging Days	15 days (2000/4/22 00:00~ 2000/5/7 00:00)
No. of Total Clients	12,005
No. of Total Queries	341,443
No. of Distinct Queries	51,125

Table 1. Some statistics of the testing proxy log.

It is noted that the recorded search queries in the log are limited to that for two representative search engine sites in Taiwan: www.kimo.com.tw and www.yam.com.tw.

In addition to identifying unique users, an effective query session segmentation algorithm has to determine which are the starting and ending requests for each user's information need. Most of search requests posse a property of time locality. Client ID with temporal information really provides a strong constraint in determining the query sessions. For this reason, we adopt an assumption similar to Silverstein et al.

that queries for a single information need come clustered in time, and then there is a gap before the user returns to the search engine.

The method for query session segmentation is then proposed as follows:

The Method for Query Session Segmentation:

For a proxy log, it will segment the whole log $L = \{T_i|$ where $T_i = (ID_k, t_i, q_i)\}$ into a set of query sessions $\{S_i| S_i = (ID_k, R_1, \ldots, R_m)$, where $R_i = (t_i, q_i)$, and $t_i - t_{i\text{-}1} <$ threshold$\}$, where t_i is the timestamp when the query q_i issued.

Analysis of Segmented Query Sessions

To realize the performance of the above method, several experiments have been performed. Fig. 3 shows the relationship between the time thresholds and the numbers of segmented query sessions. The time thresholds determine the maximum time gap between two successive requests from the same client. The values of the time thresholds were tuned from 0 seconds to 360 seconds. In the research of Silverstein et al, 5 minutes as suggested is a proper threshold value. With the same threshold value, the number of segmented query sessions is shown in Table 2. The percentages of the segmented singleton and non-singleton query sessions are found similar to those reported by Silverstein et al.

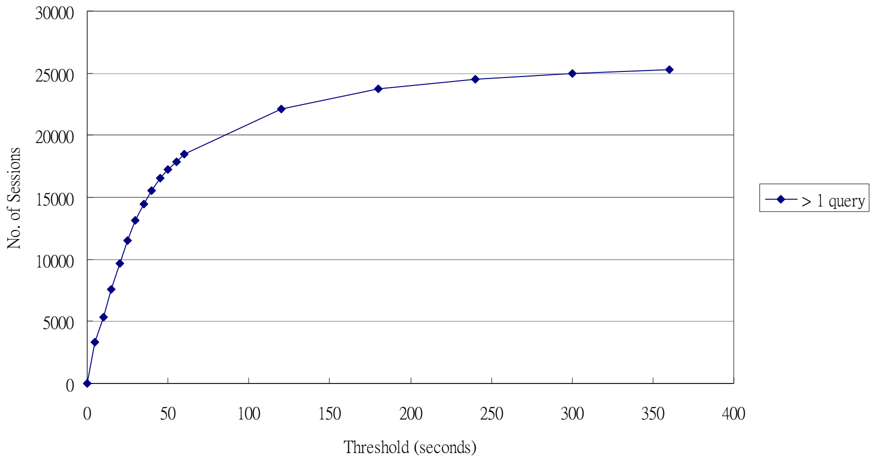

Fig 3. The numbers of segmented query sessions (that with more than 1 queries), regarding to the change of increasing time thresholds.

	No of sessions	Percentage
1 query per session	71,790	74.8%
> 2 queries per session	24,986	25.2%
Total	96,776	100%

Table 2: Percentages of the extracted singleton and non-singleton query sessions, when the time threshold is set as 5 minutes.

4. Query Session Clustering

As the definition of the session clustering problem in Section 2, the similarity estimation function is necessary and formulated below:

Similarity Estimation Between Query Sessions:

Given two sessions, S_1 = (ID_K, R_{11},…,R_{1m}) and S_1 = (ID_L, R_{21},…,R_{2n}), in which R_{ij} is the j-th query term occurred in session S_i which is issued by a client. The similarity estimation function is defined as:

$$sim(S_1, S_2) = \sum_{1<i<m, 1<j<n} sim(R_{1i}, R_{2j}) / mn$$

The similarity between two composed query terms will be further described below. Development of an effective relevance estimation function is important. Since our research is just in the beginning, only two kinds of relevance estimation functions were developed and tested. In the first method, the relevance between two query terms is simply calculated by the co-occurrence frequency value of the query terms in the segmented query sessions. In the second method, the relevance is calculated by the cosine value of the query terms' feature vectors.

Method I for Similarity Estimation of Relevant Terms

In the first method, we define the relevance estimation function below.

$$f(x, y_i) = co\text{-}occurrence(x, y_i)$$

Before calculating the relevance between query terms, a set of query sessions has been segmented and extracted from the testing proxy log. After preprocessing the query session log, we calculate the co-occurrence frequency between each unique query term and its associated terms occurring together in the same query sessions. We explain the calculation process with a simple example below. After segmenting the proxy log, it is assumed that we got five query sessions S1-5 and each contains several query terms from A to F, e.g.,

S1: {A, B}
S2: {C, D, B}
S3: {A, B, C}

S4: {A, E}

S5: {B, C, E, F}

In this case, f(B, C) will be 3, because B and C occur together in three sessions, i.e., S1, S2 and S3. Although the above method looks straightforward, its obtained performance is really out of our expectation.

Method II for Similarity Estimation of Relevant Terms

In the first method, the relevance of two query terms needs a strong support of their co-existence in a certain number of query sessions. Using a VSM-like technique it can release such a constraint. The second method is based on vector space model, and it can be formalized as below.

$f(x, y_i) = cos(FV(x), FV(y_I))$, FV(x) means feature vector of term x,

$FV(T_i) = \{S_j:N_{ij}|$ Sj and Ti are coexist in query sessions, Nij is the count of their co-occurrence}

Assuming there are two terms T_1 and T_2:

$T_1 \rightarrow \{S_1:N_{11}, S_2:N_{12}, S_4:N_{14}, S_5:N_{15}\}$

$T_2 \rightarrow \{S_1:N_{21}, S_2:N_{22}, S_3:N_{23}, S_7:N_{27}\}$

The relevance value of T1 and T2 is the obtained cosine or say the inner product value of these two vectors.

$$f(T_1,T_2)=\cos(FV(T_1)\bullet FV(T_2))=\frac{\sum_i (N_{1i}\bullet N_{2i})}{\sqrt{\sum_j N_{1j}^2}\bullet\sqrt{\sum_k N_{1k}^2}}$$

The Clustering Process

A issues to be dealt with in the clustering process, that is, what each cluster means and how to name these clusters. In order to find out the representative meaning of each cluster and avoid the difficulty in classifying short sessions, the clustering process is being developed as shown in Fig. 4, which is designed as an incremental adaptive procedure.

This procedure consists of 4 processing steps:

1. For each incoming query session, check whether there are certain common query terms between the session and existing clusters. If the common query terms exist, assign the session to these clusters.

2. If the incoming session doesn't have sufficient common query term with existing clusters, calculate the similarity between the session and existing clusters. If the estimated similarity is higher than a predefined threshold, the session will be assigned to the cluster.

3. If the incoming session isn't assigned to any cluster, it will be sent to the delay queue for further processing. In this step, the incoming session will compare with other sessions in delay queue to check whether there are common query terms in the sessions that could be combined.

4. A standalone module will dynamically merge or split the clusters according to the new requirements or the new incoming sessions. When the similarity of two clusters are higher than another pre-defined threshold, merge will happen; when the cluster grows larger, split will happen. Merging and splitting are strategies for maintaining the similarity of query sessions in a cluster.

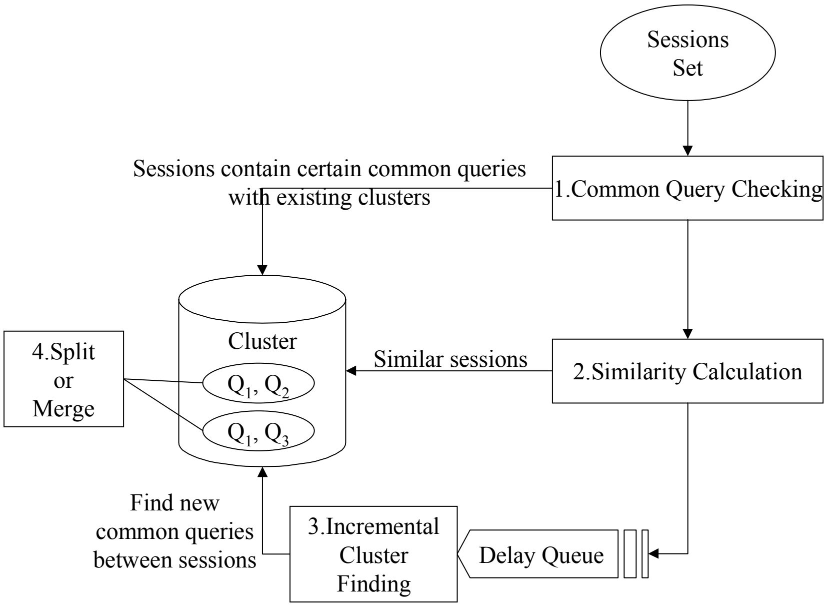
Sessions Set
1.Common Query Checking
Sessions contain certain common queries with existing clusters
2.Similarity Calculation
Similar sessions
Cluster
Q_1, Q_2
Q_1, Q_3
4.Split or Merge
Find new common queries between sessions
3.Incremental Cluster Finding
Delay Queue

Fig 4. The work flow of the session clustering process.

5. Preliminary Experiments

Performance of Query Session Clustering

The above clustering process was just implemented. The sessions grouped by Step 1 are set that should contain at least two common query terms, and each obtained cluster is then named by the pair of common query terms with the highest frequency.

Currently, there are about 700 initial clusters have been obtained from the query session log shown in Section 2. Table 3 illustrates an example of the clusters. The numbers ahead in each row of Column 2 are frequency values of the corresponding sessions. Based on our initial observations, the relevance of the clustered query sessions is often high. It is obviously higher than that obtained with document-based approach in our experiences.

Cluster Name	Sessions in Cluster
台大__台灣大學	10: 台大 台灣大學
	1: +台大 台大 台灣大學
	1: 圖書館 台大圖書館 台大 台灣大學 台灣大學圖書館
	1: 台大中國文學系 台大 台灣大學
	1: 台大 台灣大學 兵學
	1: 臺大物理 台大 台灣大學
	1: 台灣大學 台大 台灣大學^招標 台灣大學^工程 招標^台大
	1: 台大電機 台大 台灣大學
	1: 台大 台灣大學 國立台灣大學
	1: 社會學 台大圖書館 台大 台灣大學 市立圖書館
	1: 台大 台大醫學院 台大醫學系 台灣大學
	1: 時報育樂 時報育樂股份有限公司 時報 台大 台灣大學 大學聯招放榜 榜單
	1: 台灣大學 台大 成功大學
	1: 圖書 台大圖書 台大 台灣大學
	1: 椰林 台大 台灣大學 台灣大學計算機中心
	1: 台大 台灣學大 台灣大學 比賽

Table 3. An example of the obtained session clusters.

It is worthy to note that clusters with similar names (that with shared query terms as the names of the clusters) usually contain similar information needs. Table 4 is an example which contains a number of clusters with information needs related to 圖片 (picture). In these clusters, 圖片(picture) will relate to several different kinds of search subjects, including characters in cartoon (e.g. kitty and pokemon), downloading, online picture banks, greeting cards for some festivals, and etc. These similar clusters could be further taken as sub-clusters of the information needs. The obtained information would be very useful in performing term suggestion in interactive search process.

Cluster Name	Translation
卡通圖片__kitty	cartoon picture __ kitty
可愛圖片__卡通	lovable picture __ cartoon
母親節圖片__母親	mother’s day picture __ mother
母親節圖片__母親節	mother’s day picture __ mother’s day
母親節圖片__母親節卡片	mother’s day picture __ mother’s day greeting card
母親節圖片__趴趴熊	mother’s day picture __ bear
母親節圖片__康乃馨	mother’s day picture __ carnation
圖片__kitty	picture __ kitty
圖片__卡通	picture __ cartoon
圖片__卡通圖片	picture __ cartoon picture
圖片__布丁狗	picture __ pudding dog
圖片__母親節	picture __ mother’s day
圖片__母親節卡片	picture __ mother’s day greeting card
圖片__母親節圖片	picture __ mother’s day picture
圖片__皮卡丘	picture __ picachu
圖片__有趣	picture __ funny
圖片__狗	picture __ dog
圖片__趴趴熊	picture __ bear
圖片__桌面	picture __ theme
圖片__桌面王	picture __ themeking
圖片__神奇寶貝	picture __ pokemon
圖片__動物	picture __ animal
圖片__動畫	picture __ animation
圖片__康乃馨	picture __ carnation
圖片__遊戲	picture __ game
圖片__遊戲下載	picture __ game download
圖片__圖	picture __ graph
圖片__圖片下載	picture __ picture download
圖片__圖庫	picture __ picture bank
圖片__圖檔	picture __ picture file
圖片__漫畫	picture __ comic

Table 4. An example which contains a number of obtained clusters with information needs related to 圖片(picture)

Performance of Relevant Term Extraction

In fact, the proposed approach is also useful in relevant term extraction. We evaluate the proposed estimation methods with a testing set of query terms that were randomly selected from the testing proxy log. For Method I, the relevant terms are whose co-occurrence frequency large than 1, and for Method II the relevant terms are whose cosine value large than 0.25. The obtained preliminary result is shown in Table 5.

The first column “rank” in Table 5 is the order of the testing terms in the extracted term set, which is sorted by their occurrences. The real query terms are listed in the "term" column, and their English translations are listed in the next column. The data in the "freq" column represents the occurrence of each query term. The "total" column indicates the numbers of all different co-occurred terms, and the "related" column the numbers of relevant terms among co-occurred terms that were checked manually. The next nine columns are the obtained statistics of the proposed methods. Each method consists of three columns, the first is the number of extracted relevant terms, the second is the number of correct relevant terms, and the third is the obtained accuracy. Note that the third method is the result merged with the proposed two methods.

						method	1		method	2		merge		
rank	term	translation	freq	total	related	extract	related	accuracy	extract	related	accuracy	extract	related	accuracy
2	聊天室	chatroom	149	448	157	66	48	0.73	50	27	0.54	116	75	0.65
4	mp3		144	266	97	27	17	0.63	1	0	0	28	17	0.61
12	電影	movies	103	172	104	13	12	0.92	7	7	1	20	19	0.95
14	台灣大學	Taiwan U.	100	152	81	11	11	1	26	21	0.81	37	32	0.86
38	政治大學	[Univ.]	63	88	43	11	11	1	3	3	1	14	14	1
40	中國時報	Chinatimes	62	102	56	10	10	1	13	5	0.38	23	15	0.65
45	sina		60	146	46	15	13	0.87	42	14	0.33	57	27	0.47
55	pchome		52	113	48	7	6	0.86	23	1	0.04	30	7	0.23
56	華視	CTV	52	89	29	8	6	0.75	13	6	0.46	21	12	0.57
63	日本	Japan	48	92	55	6	6	1	0	0		6	6	1
111	雲林科技大學	[Univ]	34	86	54	11	11	1	38	35	0.92	49	46	0.94
116	音樂	music	34	76	46	2	1	0.5	4	4	1	6	5	0.83
204	陽明大學	[Univ]	24	35	28	4	4	1	6	5	0.83	10	9	0.9
233	司法院	Judicial Yuan	22	30	24	2	2	1	0	0		2	2	1
300	故宮	Ching Palace	17	31	19	0	0		8	8	1	8	8	1
345	證期會	[government]	16	24	21	2	2	1	5	4	0.8	7	6	0.86
531	輸入法	input method	12	14	12	2	2	1	5	4	0.8	7	6	0.86
654	九份	[Place]	10	15	11	1	1	1	4	3	0.75	5	4	0.8
760	高雄科技大學	[Univ.]	9	45	33	7	7	1	33	23	0.7	40	30	0.75
789	插圖	pictorial	9	32	10	1	0	0	16	5	0.31	17	5	0.29
818	潮州高中	[school]	9	35	5	7	0	0	22	3	0.14	29	3	0.1
884	幾米	[painter]	8	12	11	1	0	0	4	4	1	5	4	0.8
1032	宏基戲谷	[web site]	7	24	19	0	0		14	9	0.64	14	9	0.64
1092	戲劇	drama	7	20	15	0	0		7	7	1	7	7	1
1124	樂譜	music notation	7	14	13	0	0		7	7	1	7	7	1
1343	達文西特展	[Exhibition]	6	4	4	3	3	1	4	4	1	7	7	1
1629	北海道	Hokkaido	5	14	5	1	1	1	8	0	0	9	1	0.11
2220	玻璃	glass	4	19	7	0	0		14	4	0.29	14	4	0.29
2454	史記	[history book]	4	7	5	0	0		5	3	0.6	5	3	0.6
2491	圖形	graph	4	7	7	1	1	1	3	3	1	4	4	1
2515	大聯全球科技基金	[Mutual Fund]	4	3	3	1	1	1	0	0		1	1	1
2668	全華	[Publisher]	3	5	5	0	0		3	3	1	3	3	1
2900	米勒	Miller	3	6	5	0	0		3	3	1	3	3	1
3119	嶺東商專	[school]	3	8	8	1	1	1	4	4	1	5	5	1
3885	+手機	cell phone	3	5	3	0	0		2	1	0.5	2	1	0.5
4378	證券暨期貨市場發展基金會	[foundation]	2	6	6	0	0		5	5	1	5	5	1

4429	電影介紹	introduction of movie	2	2	2	0	0		0	0		0	0	
4432	格鬥	wrestling	2	7	1	0	0		6	0	0	6	0	0
4858	+智邦	[company]	2	1	1	1	1	1	0	0		1	1	1
5094	電子商務之發展	development of e-commerce	2	7	4	0	0		6	3	0.5	6	3	0.5
5274	保甄	[school addmission]	2	3	2	0	0		0	0		0	0	
5524	血管炎	endangeitis	2	2	1	0	0		0	0		0	0	
5699	藝軒	[bookstore]	2	3	2	0	0		2	1	0.5	2	1	0.5
7083	spinal^cord^compression		2	1	1	1	1	1	0	0		1	1	1
7243	木山層	[geographic term]	2	4	4	0	0		3	3	1	3	3	1
7290	小木屋	wood house	2	1	1	1	1	1	0	0		1	1	1
7563	女	female	2	7	7	0	0		6	6	1	6	6	1
8044	聯合航空公司	United Air Line	2	2	1	0	0		0	0		0	0	

Table 5. The performance obtained with the proposed methods.

Analyzing Table 5., we can find that Method I favors high frequency terms (e.g., term frequency > 50). It is really suited in applications that need not many but accurate relevant terms. However, for the query terms with not high frequency, we might rely on Method II. On the other hand, for those low frequency terms (term frequency < 10), Method II can not maintain a consistent performance. The effectiveness of this method is not reliable. In order to realize the effectiveness of the obtained result, we list an example of the extracted relevant query terms in Table 6.

The query term is 台灣大學 (Taiwan University). The obtained relevant terms can be classified into 4 major categories. The first category is abbreviations including 台大, +台大 ("+" is the query syntax). The second is synonyms with different character forms like 臺灣大學, with additional prefix like 國立台灣大學, or nick name in semantics like 椰林 (Palm trees). The third is the sub divisions of Taiwan University includes 台大醫學院 (medical school), 台灣大學計算機中心 (computing center), 台大圖書館 (library), 台大電機 (department of electrical engineering), 台大中國文學系 (department of Chinese literature), 台大醫學系 (department of medicine). The final category is the events happened in Taiwan

University, like 招標, 工程, 大學聯招放榜 and 榜單.

	Query	Frequency	
Search	台灣大學	100	
	Similar Query	Co-occur > 2	related
method 1 co-occurrence	台大	25	Y
	台大圖書館	5	Y
	台灣大學圖書館	5	Y
	+台灣大學	3	Y
	+台大	3	Y
	圖書館	3	Y
	台大醫學院	3	Y
	臺灣大學	2	Y
	國立台灣大學	2	Y
	台灣大學計算機中心	2	Y
	成功大學	2	Y
	Similar Query	Threshold > 0.25	related
method 2 VSM-like method	台大圖書	0.707	Y
	招標^台大	0.667	Y
	臺大物理	0.600	Y
	台大電機	0.600	Y
	台大中國文學系	0.600	Y
	台灣學大	0.510	Y
	台大醫學系	0.475	Y
	台灣大學^工程	0.458	Y
	台灣大學^招標	0.458	Y
	時報育樂	0.402	
	大學聯招放榜	0.402	Y
	時報育樂股份有限公司	0.402	
	社會學	0.397	Y
	時報	0.328	
	比賽	0.312	
	椰林	0.305	Y
	台大醫學院圖書館	0.278	Y
	榜單	0.272	Y
	市立圖書館	0.262	

Table 6. An example of relevant terms extracted with the proposed methods.

For more references, there are several examples that were not used in the testing are also illustrated below:

- 手機 76
    - 台灣大哥大:6 中華電信:5 易利信:4 T10:2 諾基亞:2 安瑟:2 桌面王:2 行動電話:2 motorola:2 中古手機:2 全虹:2 sagem:2
- 圖片 45
    - 皮卡丘:7 卡通圖片:7 神奇寶貝:4 圖庫:4 圖:3 kitty:3 聊天室:3 漫畫:3 趴趴熊:3 皮卡丘圖片:2 小番薯:2 美麗人生:2 母親節:2 美女:2 動畫:2 康乃馨:2 日本卡通:2 圖畫:2 桌面王:2 布丁狗:2

- 圖庫 41
  - 圖片:4 動畫:3 圖檔:3 網頁製作:2 圖:2 網頁圖庫:2 遊戲下載:2 世界地圖:2 地圖:2
- 替代役 20
  - 國防部:5 社會役:3 兵役:3 南投縣政府:2 內政部:2 國防役:2

6. Conclusion

In this paper, a new approach based on log analysis is proposed for implementing interactive Web search. The most important feature of the proposed approach is that the suggested terms corresponding to a user query are extracted from similar query sessions, rather than from the contents of the retrieved documents. Furthermore, the estimation of term relevance is also based on co-occurrence analysis of the query terms in query sessions. The experiment results presented in this paper are based on analysis of the proxy server logs. The results obtained so far demonstrate that the proposed approach is quite promising in respect to improving the effectiveness of interactive web search engines.

The results presented in this paper is just a beginning of mining log data toward developing more effective web search engines. Since this approach already demonstrates quite promising results, further investigation on mining log data deserves more of our attention. Further study may result in more advanced mining mechanism that can give us more comprehensive information about term relevance and allow us to identify users' information need more effectively. For example, some sort of thesaurus information may be derived from mining log data.

References

[1] AltaVista. http://www.altavista.com.

[2] P.G. Anick and S. Tipirneni, “The Paraphrase Search Assistant: Terminology Feedback for Iterative Information Seeking,” in Proceedings of 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99), pages 153-159, 1999.

[3] E.F. de Lima and J.O. Pedersen, “Phrase recognition and expansion for short precision-biased queries based on a query log,” in Proceedings of 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99), pages 145-152, 1999.

[4] Direct Hit. http://www.directhit.com.

[5] Infoseek. http://www.infoseek.com.

[6] B.J. Jansen, A. Spink, J. Bateman, and T. Saracevic, “Real life information retrieval: A study of user queries on the web,” SIGIR FORUM, 32(1), 1998.

[7] S. Jones and M.S. Staveley, “Phrasier: a System for Interactive Document Retrieval Using Keyphrases,” in Proceedings of 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99), pages 160-167, 1999.

[8] Kimo. http://www.kimo.com.tw.

[9] C. Silverstein, M. Henzinger, H. Marais, and M. Morics., “Analysis of a very large AltaVista query log,” Technical Report 1998-014, Digital Systems Research Center, 1998.

[10] B. Velez, R. Weiss, M.A. Sheldon and D.K. Gifford, “Fast and Effective Query Refinement,” in Proceedings of 20th International Conference on Research and Development in Information Retrieval (SIGIR-97), pages 6-15, 1997.

[11] E. Voorhees and D.K. Harman, "Overviw of the sixth text retrieval conference TREC-5," in Proceedings of the Fifth Text REtrieval Conference (TREC-5), 1997

[12] E. Voorhees and D.K. Harman, "Overviw of the sixth text retrieval conference TREC-6," in Proceedings of the Sixth Text REtrieval Conference (TREC-6), 1998

[13] D. Wessels, SQUID Frequently Asked Questions. Section 6. Squid Log Files. http://www.squid-cache.org/Doc/FAQ/FAQ-6.html

[14] Yam. http://www.yam.com.tw.