ISSN:0975-9646

Sami Uddin et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6744-6749

Recent Trends in Users' Query Clustering

Sami Uddin , Amit kumar Nandandwar

CSE Department
VNS College, Bhopal, India

Abstract— **Lots of information are gathered in dataminings that are navigated and explored for analytical functions. Solely arise the matter of recommending a dataminings query to a dataminings user attracted attention. During this paper, we tend to gives an easy formal framework for expressing dataminings query recommendations. Search engines are programs that search documents for nominal keywords and come back an inventory of the documents wherever the keywords were found. They come back long list of hierarchic pages, finding the relevant info associated with a specific topic is changing into progressively essential and so, Search Result optimization techniques are available to play. During this work Associate in nursing formula has been applied to suggest connected queries to a query submitted by user. Query logs are vital info repositories to stay track of user activities through the search results. Query logs contain attributes like query name, clicked uniform resource locator, rank, time. Then the similarity supported Keyword and Clicked URL's is calculated.**

Keywords— **Put your keywords here, keywords are separated by comma.**

I. INTRODUCTION

Clustering of program queries has attracted vital attention in recent years. Several program applications like query recommendation need query agglomeration as a pre-requisite to operate properly. Indeed, agglomeration is critical to unlock verity price of query logs. However, agglomeration search queries effectively are kind of difficult, owing to the high diversity and capricious input by users. Search queries are sometimes short and ambiguous in terms of user necessities. Many alternative queries might confer with one construct, whereas one query might cowl several ideas. Existing current agglomeration ways, like K-Means or DBSCAN cannot assure sensible leads to such a various atmosphere. Clustered agglomeration offers sensible results however is computationally quite pricey. This paper presents a unique agglomeration approach supported a key insight – program results may themselves be wont to determine query similarity. This work proposes query matter similarity and time thresholds for a lot of strong approach that leverages search query logs. This can be wont to develop a awfully economical and correct formula for agglomeration queries. This system can useful for varied search engines and search applications like query suggestions, result ranking, query alterations, sessionization, and cooperative search.

With the event in data technology, the net [1] has clad to be a huge data repository covering virtually each space, within which an individual's user may be concerned. In spite of recent advances in net program technologies, there ar still several things within which user is bestowed with unwanted and non- relevant pages within the high most results of the hierarchal list. Program usually has difficulties in forming a pithy and precise illustration of the response pages appreciates a user query. Providing a group of websites supported user query words isn't an enormous drawback in program. The problem arises at the user finish as he must sift through the long result list, to search out his desired content. This drawback is remarked as data Overkill drawback [2].

The design [3] of the program is shown in Figure1.

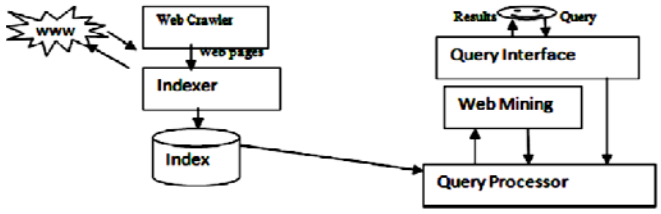

Figure 1: Architecture of Search Engine

There are three parts in programmed referred to as Crawler, skilled worker and Ranking mechanism. The crawler is additionally known as mechanism that navigates the online and downloads the online pages. The downloaded pages area unit transferred to associate classification module and erect the index supported the keywords in individual pages. once a query is being floated by a user, it means that the query transferred in terms of keywords on the interface of a quest engine, the query mainframe section examine the query keywords with the index and precedes the URL's of the pages to the shopper. However gifting the pages to the shopper a ranking mechanism is completed by the search engines to present the foremost relevant pages at the highest and fewer vital pages at all-time low.

As of nowadays, the indexed net contains a minimum of thirty billion pages [3]. In fact, the general net could comprise over one trillion distinctive URLs, a lot of and a lot of that is being indexed by search engines each day. Out of this quagmire of knowledge, users usually rummage around for the relevant info that they require by move search queries to go looking engines. The matter that the search engines face is that the queries area unit terribly various and sometimes quite imprecise and/or ambiguous in terms of user needs. Many various queries could ask one thought, whereas one query could correspond to several ideas. To arrange and convey some order to the current huge unstructured dataset, search engines cluster these queries to cluster similar things along. To extend usability, most business search engines, like Google, Yahoo!, Bing, and raise additionally augment their search facility through extra services like query recommendation or query suggestion. These services create it a lot of convenient for users to issue queries and acquire correct results from the

programmed, and so area unit quite valuable. From the programmed perspective, effective clump of search queries could be a necessary pre-requisite for these services to perform well. As a result of all of those reasons, clump of program queries has attracted vital attention in recent years. However, existing current clump strategies, like K-Means or DBSCAN cannot assure smart ends up in such a various atmosphere. There area unit many challenges expose by the distinctive nature of the atmosphere. The first issue is to work out the way to live similarity between queries. To change a lot of precise info retrieval, a representative and correct descriptor is indispensable for computing the similarity between queries.

The thought of query similarity was originally utilized in info retrieval studies [4]: measurement the similarity between the content-based keywords of 2 queries. However, the matter with victimization this within the query log atmosphere is that users' search interests aren't continuously identical although the issued queries contain identical keywords. for example, the keyword "Apple" could represent a well-liked quite fruit whereas it's additionally the keyword of a well-liked company "Apple Iraqi National Congress.". Hence, the utilization of content-based keywords descriptor is very restricted for this purpose.

A. Query Logs

The log keeps user's queries and their clicks further as their browsing activities. The standard logs [5] of program embody the subsequent entries:

1) User IDs
2) Query Q issued by the user
3) Address u clicked by the user
4) Rank r of the address u clicked for the query Q
5) Time t at that the query has been submitted

The information contained in query logs may be utilized in many ways [6, 7], example to produce context throughout search, to classify queries. Query log is shown in Table one.

TABLE 1: QUERY LOGS

User Id	*Query Clicked*	*URL*	*r*	*Time*
Admin	*Data Mining*	*www.dming.com*	*6*	*12:10*
Admin	*Data ware housing*	*www.dming.com*	*5*	*8:30*
Admin	*Data Mining*	*www.google.com*	*5*	*11:10*

In this paper, we tend to survey the prevailing strategies for computing query recommendations. We tend to prohibit the scope of this survey to strategies that, given a user's query, use it or rework it into another query, with a supposed intercalary price for the user's exploration. We tend to propose a proper definition of this downside, specifically to visualize the advice of queries for exploration functions as a recommending perform A survey of query recommendation techniques for exploration taking as input: The query log, a specific query session known as the present session, a user profile, a instance, associated an expectation perform. Given these parameters this recommending perform outputs a group of suggested queries, every with a given rating indicating the interest of the query for the present session.

Subsequently, to live similarity between 2 queries, the query illustration of a vector of URLs in a very clickthrough bipartite graph [8][9] has been adopted. even so, despite however giant the query log information set is, it's doable that the entire search intent of some queries might not be adequately portrayed by the obtainable click-through info. for example, in a very specific large-scale query log, there is also no clicked address for the query "Honda vs Toyota". Therefore, although it's clearly relevant to the query "Honda", on the idea of this click-through information, there's no similarity. Therefore, existing query log information isn't correct enough for analysing users' search intent, particularly for those queries with none clicked address. One more reason that causes quality is that the query log information comprises users' click-through info in a very specific amount, whereas search interests may even modification over time. If we tend to utilize associate aggregative query logs collected in a very long amount to check and cluster queries, the accuracy is also wedged.

II. RELATED WORK

Therefore, describing queries solely by content-based keywords or strictly through click-through knowledge isn't forever correct for program query clump. During this paper, our main contribution is to propose a completely unique query descriptor for scrutiny queries. this is often supported a key insight – program results may themselves be wont to establish query similarity; and so incorporate each content and click on through info [11][12][13][14]. Supported this, we tend to additionally outline a replacement similarity metric which will be utilized in any distance primarily based clump algorithmic rule. owing to the variety of queries and therefore the curse of spatiality, current clump algorithms have high machine price. we tend to additionally propose AN economical clump algorithmic rule to scale back machine price. we tend to compare the query clump results of our approach with many existing state of the art strategies and show that our algorithmic rule provides smart cohesion, separation on clustered queries and considerably reduced runtime.

Query clump has its roots in keywords-based info retrieval analysis [4]. Since most of the keywords area unit ambiguous, analysing the content of query keywords or phrases by ancient info retrieval techniques has several limitations. Following that, click-through query logs are mined to yield similar queries [20]. Beeferman and Berger [8] 1st introduced the agglomerate clump methodology to get similar queries mistreatment query logs however with limitations (noise and little variety of common clicks). The query clump approach adopted in [21] uses a K-Means clump approach. K-Means algorithmic rule cannot adapt well in query clump case owing to the issue on specifying k. Wen et al. [22] analyzed each query contents and clickthrough bipartite graph and applied a density-based algorithmic rule DBSCAN [23] to cluster similar queries. The same as agglomerate query clump, DBSCAN algorithmic rule adopted in [22] needs high computation price. Meanwhile, Wen et al. linearly mix measures on content-based similarity and cross-references primarily based similarity however it's tough to line parameters for

linear combination of 2 similarity metrics. However, our hierarchical search results knowledge (enforced by [11][12][13][14]) naturally contemplate each factors. Moreover, we've got compared the accuracy of our approach with the geometer distance primarily based query log clump [10]. Moreover, Kendall's letter [16] and a few relevant measures [24] are often effective in mensuration the accuracy of clustered queries. Since our most vital contribution is in observant the very fact that search results are often wont to perform query clump, they could be effective metrics additionally. However, since search queries clump may be a acknowledge exhausting downside, there's no economical algorithmic rule that has been projected for clump queries with reference to existing top-k lists scrutiny measures.

Finally, tend to use profile constant to live the cohesion and separation of query clump results. Larger profile constant indicates giant inter-cluster distances and little intra-cluster distances. Davies-Bouldin index [25][26] seeks identical objective on clump validation. the excellence between silhouette constant is that – decreased index generates the simplest clump results.

III. Information Used For Query Process

Although search and browse log knowledge offer nice opportunities for enhancing internet search, there are many challenges before such knowledge are often utilized in varied applications. First, the scale of log knowledge is sometimes terribly giant. In observe the scale of search and browse log knowledge at an exploration engine is usually at the magnitude of tens of terabytes day after day. Second, log knowledge area unit quite shouting. For instance, queries could also be issued by machines for experiments; user input in computer address boxes could also be redirected to look engines by internet browsers; and clicks on search result pages could also be arbitrarily created by users.

To overcome noise and volume, one will combination raw log knowledge in pre-processing. By summarizing common patterns in data, the scale of information is often greatly reduced. Moreover, when aggregation, we tend to could prune patterns with low frequencies to scale back noise.

One query is a way to summarize raw log knowledge for varied log mining tasks. In fact, search and browse log knowledge have terribly complicated knowledge structures with varied sorts of knowledge objects and relationships. the info objects could embody users, sessions, queries, search result pages, clicks on search results, and follow-up clicks. These differing types of information objects kind a hierarchy. At the highest level, every user incorporates a series of sessions, wherever every session contains a sequence of queries. In a query, a user could open many WebPages. Finally, a user could additional follow the hyperlinks within the WebPages of search results and browse additional WebPages. Additionally to the hierarchal relationship between differing types of information objects, the info objects at identical level typically kind a successive relationship. Here, we tend to introduce four sorts of knowledge account that area unit wide utilized in log mining, namely, query histograms, click-through bipartite, click patterns, and session patterns. Among the literature reviewed during this survey, ninetieth of the papers on log mining utilized a minimum of one in all the four sorts of knowledge account.

A. Query bar graph

A query bar graph represents the quantity of times every query is submitted to an exploration engine. As shown in Figure three, query bar graph contains query strings and their frequencies. As an easy statistics, query bar graph are often utilized in a good form of applications, like query motor vehicle completion and query suggestion.

B. Click-through Bipartite

A click-through bipartite graph, like Figure a pair of, summarizes click relations between queries and URLs in searches. The bipartite graph consists of a collection of query nodes and a collection of computer address nodes. A query and a computer address area unit connected by a position if the computer address is clicked by a user once it's came back as a solution to the query. A weight cij could also be related to AN edge eij, indicating the full variety of times URL uj is clicked with reference to query qi. Click-through bipartite is maybe the foremost wide used organization in log mining. As we are going to see within the following sections, it are often used for query transformation, query classification, document annotation, and plenty of alternative tasks.

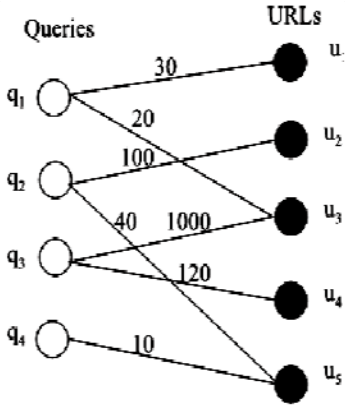

Figure. 2. An example of click-through bipartite graph. In a click-through bipartite graph, nodes represent queries and URLs, and edges represent click relations between queries and URLs.

C. Click Patterns

Click patterns summarize positions of clicked URLs in search results of queries. To be specific, every search result (also called search impression) Iq with respect to query q are often portrayed by Iq=(q;L), wherever L may be a list of triples (u,p,c), wherever u is that the universal resource locator of a page, p is that the position of the page, and c indicates whether or not the page is clicked. The identical search results are more collective to 1 click pattern Pq =(q;L;cc), wherever cc is that the variety of search results, samples of click patterns. In follow, an inventory L solely includes the highest N URLs. Compared with a click-through bipartite, click patterns contain richer info. A click-through bipartite solely represents collective clicks of URLs,

whereas click patterns more represent the positions of the clicked URLs furthermore as unclicked URLs. As are going to be seen within the later sections, click patterns will facilitate several tasks in search, like classifying steering and informational queries, learning pairwise document preference, building ordered click models, and predicting user satisfaction.

D. Session Patterns

Session patterns summarize transitions among queries, clicks, and browses at intervals search sessions. In fact, session patterns are often outlined in several ways in which looking on specific applications. As an example, sequences of queries as sessions and extract frequent query sequences as session patterns. In different cases, session patterns might involve not solely queries however additionally clicked URLs. as an example, Cao et al. [2009] outlined session patterns supported sequences of queries and their clicked URLs. Since session patterns represent users' search behaviours in an exceedingly additional precise means, it's been used extensively. As are going to be seen later, session patterns are wide utilized in tasks like query transformation, document ranking, and user satisfaction prediction.

IV. Recent Query Clustering Techniques

A. Extracting User Interests from Search query Logs

The paper [22] proposes to reinforce search query log analysis by taking into consideration the linguistics properties of query terms. we have a tendency to initial describe a technique for extracting a world linguistics illustration of a quest query log and so show however we are able to use it to semantically extract the user interests. The worldwide illustration consists of a taxonomy that organizes query terms supported generalization/specialization ("is a") linguistics relations and of operate to live the linguistics distance between terms. It has a tendency to then outline a query terms clump rule that's applied to the log illustration to extract user interests. The analysis has been done on massive real-life logs of a well-liked program.

B. Query results Cache supported log analysis

Query results cache is an efficient technology to boost the performance of net search engines. during the paper [23], have a tendency to conducted associate analysis of net method query logs from SOGOU INC. to explore some problems with query results cache for net scheme as well as the neighbourhood of net search engine employment and also the impact of cache replace policy. it also has a tendency to additionally conducted a series of experiments on query results cache with a spread of cache capability and cache replacement policy settings. Experimental results showed that once there are solely a little variety of historical query logs, it might be higher to decide on a dynamic policy, and hybrid cache approach had important improvement than the other policies in smaller cache capability.

C. Query-Based Clusters and Labels

Current net search engines, like Google, Bing, and Yahoo!, rank the set of documents S retrieved in response to a user query and show the universal resource locator of every document D in S with a title and a snipping, that is associate abstract of D. Snippets, however, don't seem to be as helpful as they're designed for, that is meant to help its users to quickly establish results of interest, if they exist. These snippets fail to (i) offer distinct info and (ii) capture the most contents of the corresponding documents. Moreover, once the supposed info would like laid out in a quest query is ambiguous, it's terribly troublesome, if not not possible, for a quest engine to spot exactly the set of documents that satisfy the user's supposed request while not requiring extra inputs. Moreover, a document title isn't continuously an honest indicator of the content of the corresponding document. All of those style issues are often resolved by paper [24] planned query-based cluster and labeller, called QCL. QCL generates summary clusters of documents covering numerous subject areas retrieved in response to a user query that saves the user's time and energy in sorting out specific info of interest while not having to flick through the documents one by one. Experimental results show that QCL is effective and economical in generating high-quality clusters of documents on specific topics with informative labels.

D. Query process in Metric-Space Similarity

Metric-space similarity search has been verified appropriate for looking out massive collections of advanced objects like pictures. Variety of distributed index knowledge structures and individual parallel query process algorithms are planned for clusters of distributed memory processors. Previous work has shown that best performance is achieved once mistreatment world classification as against native classification. But world classification is at risk of performance degradation once query load becomes unbalanced across processors. This paper [25] proposes a query programming rule that solves this downside. It adaptively load balances process of user queries that are dynamically skew towards explicit sections of the distributed index. Sections extremely hit by queries are often unbroken replicated. Experimental results show that with 1%-10% replication performance improves considerably (e.g., 35%) underneath skew work-loads.

E. Improving Delay in camera net Search

[26] The net may be a vast repository of knowledge. Net search engines are a basic tool for locating and accessing all this info. However, these tools may also be a threat for the privacy of their users. This happens as a result of users of t times reveal non-public info in their queries. Net search engines gather this personal knowledge, store it throughout an outsized amount of your time and use it to boost their search results and to extend their economical advantages. So as to avoid this case, it's necessary to supply net search ways that preserve the privacy of the users. Current proposals within the literature increase considerably the query delay. This is often the time that users have to be compelled to wait so as to get the search results for his or

her queries. During paper [26], we have a tendency to propose a modification of the Useless User Profile (UUP) protocol. The ensuing theme has been tested in associate open setting and also the results show that it achieves rock bottom query delay that has been rumoured within the literature. Additionally thereto, it incentivizes users to follow the protocol so as to safeguard their privacy.

F. Query clump mistreatment Top-k Search Results

Search queries are typically short and ambiguous in terms of user needs. Many alternative queries might talk over with one idea, whereas one query might cowl several ideas. Existing current clump ways, like K-Means or DBSCAN cannot assure sensible leads to such a various setting. Collective clump offers sensible results however are computationally quite expensive. This paper presents a unique clump approach supported a key insight – method results may themselves be wont to establish query similarity. Paper [27] have a tendency to propose a unique similarity metric for various queries supported the hierarchal universal resource locator results came by a quest engine for queries. This is often wont to develop an awfully economical and correct rule for clump queries.

G. Location-Based Search Queries

[28] Location-based applications utilize the positioning capabilities of a mobile device to work out this location of a user, and customize query results to incorporate neighbouring points of interests. However, location data is usually perceived as personal info. one in every of the immediate problems clogging the wide acceptance of location-based applications is that the lack of acceptable methodologies that provide fine grain privacy controls to a user while not immensely touching the usability of the service. Whereas varieties of privacy-preserving models and algorithms have taken form within the past few years, there ought to associate nearly universal to specify one's privacy demand while not understanding its implications on the service quality. Paper [28] propose a user-centric location-based service design wherever a user will observe the impact of location quality on the service accuracy before deciding the geo-coordinates to use in an exceedingly query. we have a tendency to construct an area search application supported this design and demonstrate however meaty info are often changed between the user and also the service supplier to permit the reasoning of contours representational process the amendment in query results across a geographical area. Results indicate the likelihood of enormous default privacy regions (areas of no amendment in result set) in such applications.

H. Pair wise Learning to Rank for Search query

This text introduces a brand new rule for a quest query writing system Correction System. It's supported learning to rank approach associated permits mistreatment sizable amount of varied signals resulting in an improved accuracy. The performance is going to be tested against the traditional resolution - the abuzz Channel Model. The new system was developed on a Czech net search query set, however the feature vector structure and also the rule are often simply custom-made for the other language once comfortable knowledge is out there. Authors in [29] have described the rule details, the coaching and validation knowledge sets. Further, we are going to discuss the choice and impact of the new feature vector signals.

TABLE 2
COMPARISON OF CLUSTERING PROTOCOLS

Author	Year	Model	Processing techniques	Application
Limam, L. [22]	2010	enhance s earch que ry log analysis	extracting a global semantic representation of a search query log	real-life logs of a popular searc h engine
Hongyu an Ma [23]	2013	Cache based on log analysis	search engine workload and the impact of cache replace policy	improve the performance of Web search engines
Qumsiy eh, R. [24]	2013	Clusters and Labeler	clusters of documents covering various subject	high-quality clusters of documents
Gil-Costa, V. [25]	2012	Metric-space similarity search	load balances processing of user queries	searching large collections of images
Romero -Tris, C. [26]	2011	query delay	Useless User Profile (UUP) protocol	Web search engines
Yuan Hong, [27]	2011	novel similarity metric	diverse queries based on the ranked URL results returned by a search engine	Search queries clustering
Dewri, R. [28]	2014	neighbori ng points of interests	utilize the positioning capabilities of a mobile device to determine the current location	Service provider to allow query results across a geographic area.
Novak, A. [29]	2013	Search Query Spelling Correctio n System	learning to rank approach	selection and impact of the new feature vector

V. CONCLUSION

In this article, given a survey on search and browse log mining for net search, with the main focus on up the effectiveness of net search by query understanding, document understanding, document ranking, user understanding, and observation and feedback. As reviewed, several advanced techniques were developed. Those techniques were applied to large amounts of search and browse log information out there at net search engines and were powerful in enhancing the standard of the search engines. A query log is formed that contains attributes like query name, Clicked uniform resource locator, rank, time. By victimization query log it'll calculate similarities primarily based on keywords and clicked url's by victimization formulas given in paper. Then it will calculate combined similarity and cluster the queries on the premise of threshold worth provided by user. To extract most worth from search queries, search engines should develop economical approaches for generating additional precise and multi-functional clusters of comparable queries. However, most current cluster ways all suffer from bound limitations on cluster this extremely various set of queries.

There are still several difficult and attention-grabbing issues for future work. we tend to list 3 of them here as examples. First, it's difficult to affect the long tail in search and browsing log effectively. Search and browse log information are user behaviour information and follow the ability law distributions in several aspects. Sometimes it's simple to mine helpful information from the top a part of an influence law distribution. a way to propagate the well-mined information from the top half to the tail half continues to be a challenge for many log mining tasks. Second, it's necessary to leverage different data or information in mining. Log mining chiefly focuses on the employment of log information. it might be useful to leverage data or information in different information sources throughout the mining method, like Wikipedia. it's necessary to conduct additional analysis on log mining in such a setting.

REFERENCES

[1] A. Arasu, J. Cho, H. Gracia-olina, A. Paepcke, and S. Raghavan, "Searching the Web", ACM Transactions on Internet Technology , Vol.1 No. 1, pp.97-101, 2001.

[2] A. Borchers, J. Herlocker, J. Konstanand, and J. Riedl,"Ganging up on information overload", computer, vol. 31, No. 4, pp. 106-108, 1998.

[3] "http://www.worldwidewebsize.com/."

[4] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, Inc., 1986.

[5] Edgar Meij, Marc Bron, Bouke Huurnink, Laura Hollink, and Maarten de Rijke. Learning semantic query suggestions. In 8th International Semantic Web Conference (ISWC 2009). Springer,October 2009.

[6] K. Hofmann, M. de Rijke, B. Huurnink, E. Meij. A Semantic Perspective on Query Log Analysis, In Working notes for the CLEF 2009 Workshop, Cortu, Greece.

[7] H. Ma, H. Yang, I. King, and M. R. Lyu. Learning latent semantic relations from click through data for query suggestion. In CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management, pages 709–718, New York, NY, USA, 2008. ACM.

[8] U. Irmak, V. von Brzeski, and R. Kraft, "Contextual ranking of keywords using click data," inICDE, pp. 457–468, 2009.

[9] F. Radlinski and T. Joachims, "Query chains: learning to rank from implicit feedback," in KDD, pp. 239–248, 2005.

[10] T. Joachims, L. A. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, "Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search," ACM Trans. Inf. Syst., vol. 25, no. 2, 2007.

[11] R. Fagin, R. Kumar, and D. Sivakumar, "Comparing top k lists," SIAM J. Discrete Math., vol. 17, no. 1, pp. 134–160, 2003.

[12] C. H. Papadimitriou and K. Steiglitz, Combinatorial optimization: algorithms and complexity. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1982.

[13] M. Barbaro and T. Z. Jr., "A face is exposed for aol searcher no. 4417749," Augest 9, 2006. (New York Times).

[14] K. Hafner, "Researchers yearn to use aol logs, but they hesitate," Augest 23, 2006. (New York Times).

[15] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, "Query expansion by mining user logs,"IEEE Trans. Knowl. Data Eng., vol. 15, no. 4, pp. 829–839, 2003.

[16] R. Baeza-Yates, C. Hurtado, and M. Mendoza, "Query recommendation using query logs in search engines," inEDBT, 2004.

[17] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Clustering user queries of a search engine," inWWW '01.

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A densitybased algorithm for discovering clusters in large spatial databases with noise," inKDD, pp. 226–231, 1996.

[19] E. Yilmaz, J. A. Aslam, and S. Robertson, "A new rank correlation coefficient for information retrieval," in SIGIR, pp. 587–594, 2008.

[20] D. L. Davies and D. W. Bouldin, "A cluster separation measure,"IEEE Transactions on In Pattern Analysis and Machine Intelligence, vol. PAMI-1, pp. 224–227, Nov. 1977.

[21] S. Saitta, B. Raphael, and I. F. C. Smith, "A bounded index for cluster validity," in MLDM, pp. 174–187, 2007.

[22] Limam, L. ; Coquil, D. ; Kosch, Harald ; Brunie, L., "Extracting User Interests from Search Query Logs: A Clustering Approach", Database and Expert Systems Applications (DEXA), , Page(s): 5-9, IEEE, 2010

[23] Hongyuan Ma, "Research on query results Cache based on log analysis in web search engines", 3rd International Conference on Consumer Electronics, Communications and Networks (CECNet), Page(s): 551-554, IEEE, 2013.

[24] Qumsiyeh, R. ; Yiu-Kai Ng, "Enhancing Web Search Using Query-Based Clusters and Labels" IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Volume:1, Page(s): 159 -164, IEEE, 2013.

[25] Gil-Costa, V. ; Marin, M., "Load Balancing Query Processing in Metric-Space Similarity Search", International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM Page(s): 368-375, IEEE, 2012.

[26] Romero-Tris, C. ; Viejo, A. ; Castella-Roca, J., "Improving Query Delay in Private Web Search" International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Page(s): 200-206, IEEE, 2011.

[27] Yuan Hong, Jaideep Vaidya and Haibing Lu, "Search Engine Query Clustering using Top-k Search Results", International Conferences on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM, 2011.

[28] Dewri, R. ; Thurimella, R., "Exploiting Service Similarity for Privacy in Location-Based Search Queries", IEEE Transactions on Parallel and Distributed Systems, Volume: 25, Issue: 2, Page(s): 374-383, IEEE, 2014.

[29] Novak, A. ; Sedivy, J., "Pairwise Learning to Rank for Search Query Correction", International Conference on Systems, Man, and Cybernetics (SMC), Page(s): 3054 - 3059, IEEE, 2013.