International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Efficient Query Suggestion System using User Search Log

Prajakta Shinde[Á*] and Pranjali Joshi[Á]

[Á]Computer Department, Pune University, PICT,Pune India

Accepted 16 July 2014, Available online 01 Aug 2014, **Vol.4, No.4 (Aug 2014)**

Abstract

With the growing information burst on the World Wide Web, internet has placed high demands on search engines. Existing search engines provide most of the features for user query. But users of internet are not satisfied with them as they return thousands of documents in response to user query. So to develop user search intent application is challenging, satisfying increased expectations & diverse needs of user. Recorded user search logs are analyzed and used to form clusters. This clustering of user query is imperative for filtering the relevant results, so in the proposed system first approach mines frequent query patterns from users search history using FP growth, if user want to select any query from his/her previous search history then he can. Otherwise he will enter new query and second approach identifies clusters of queries from all users search history those clusters of queries which is similar to current query are query suggestions. Thus, by automating the optimization process of searching on web; we can minimize user efforts; maximize user satisfaction for getting desired search.

Keywords: *Information search and retrieval, frequent item set mining, query clustering, search history, suggestion*.

1. Introduction

With the explosive growth of web data, organizing and utilizing information effectively has become more important. So clustering of query log and query suggestion techniques has made significant progress to use web data effectively. There are many ways for clustering users search history, such as agglomerative clustering, clustering based on query flow graph and so on explained in next section .These methods of clustering have major drawbacks because temporal and textual features are considered. Although the time-based and text-based relevance metrics may work well in some cases. For time based metric, one can assume that a query is always followed by a related query; however, this may not be the case when the user is multitasking. Similarly, the text-based metrics such as jaccard similarity and cosine similarity can capture the relevance between query groups around textually similar queries such as iPod and apple iPod but will fail to identify relevant query groups around queries such as iPod and apple store since they are not textually similar. Additionally, the text-based metrics may mistakenly identify query groups around, say, jaguar car manufacturer and jaguar animal reserve as relevant, since they share some common text.

Therefore, we need a relevance measure that is robust enough to identify similar query groups beyond the approaches that simply rely on the textual content of queries or time interval between them (Hwang *et al* 2012). Proposed system makes use of search logs in order to determine the relevance between query groups more effectively. In fact, the search history of a large number of users contains metrics regarding query relevance, such as queries tend to be issued closely together, and which queries tend to lead to clicks on similar URLs . Such metrics are user generated and are likely to be more robust, especially when considered at scale. We suggest measuring the relevance between query groups by exploiting the query logs and the click logs simultaneously, and mining search logs using FP growth algorithm. In the proposed system the first approach mines frequent query patterns from users search history using FP growth, and the second approach identifies clusters of queries from all users search history those cluster of queries which is similar to current query are query suggestions.

The paper is organized as follows. Section I gives introduction, Section II gives the review of user search log techniques that is related work, section III presents mathematical model of system in section IV flow of proposed system, section V presents experimental results of system and in last section conclusion is presented.

2. Related Work

Various approaches have been proposed in recent years that use query logs for query suggestion.

(Beeferman 2000) apply a hierarchical agglomerative clustering technique to click-through data to find clusters of similar queries and similar URLs in a Lycos log. A bipartite graph is constructed from queries and related URLs which is iteratively clustered by choosing at each iteration the two pairs of most similar queries and URLs; but with limitations of noise and small number of common clicks

*Corresponding author: **Prajakta Shinde**

The query clustering approach in (Baeza-Yates *et al* 2004) uses K-Means clustering algorithm but which cannot work that much effectively in query clustering case due to the difficulty on specifying value of k.

Wen *et al*. (J Wen *et al* 2002) analyzed query contents as well as click through bipartite graph and applied a density-based algorithm

DBSCAN (M Ester *et al* 1996) to form cluster of similar queries. Similar to agglomerative query clustering, DBSCAN algorithm requires high computation cost.

In (Boldi *et al* 2008) graph representation of the interesting knowledge about latent querying behavior is done using query flow graph. In the query-flow graph a directed edge from query qi to query qj means that the two queries are likely to be part of the same search mission. Time and textual properties are considered for grouping so it is not that much efficient. So In (Harksoo *et al* 2008) for FAQ retrieval clustering of query log is done using latent term weights.

Jeonghee Y (Maghoul *et al* 2009) introduced click through graph which consider query and clicked page relationship. Query clustering is done on the basis of Query and clicked page relationship, other features are not taken into account.

Ji-Rong Wen (Ji Rong 2001) introduced query clustering approach using content words and user feedback, Combining Content and feedback similarity approach so it is efficient but it's difficult to set parameters for linear combination of two similarity metrics.

Yuan Hung, Jaideep V (Yuan Hung 2011) and (Kajal Y *et al* 2011) used search results for query clustering, Similarity based on ranked url results return by search engine this approach is having better scalability.

(Toru Onada *et al* 2008) introduced concept of query clustering based on history of query frequency, but its limitation is it is applicable for short terms only.

(Lye Limam 2000) they applied semantic taxonomy to search log and perform grouping to extract user interest which is helpful for personalization application.

In this paper, we are going to give the idea about query clustering which go beyond the beyond approaches that rely on textual similarity or time thresholds; and query suggestion is given from the cluster which will match with current query.

3. Mathematical Model

The mathematical model for the proposed system is stated below.

Objective: To provide facility of efficient query suggestion using user search log.

Let S be the system, such that

S = {I, O, F, Su, Fa}
I = Input to the system
O = Output of the system
F = Set of functions
Su = Success
Fa = Failure

Input

I = User search history

= {D1, D2…Dn} Di= user data, $1 \leq i \leq n$

Output

O = {List of best query suggestions for current query}.

= $\{q_1, q_2, q_3, \ldots\}$ qk=query suggestions $1 \leq i \leq k$

F1– Function to find frequent queries by applying FP growth algorithm [5] [2] [9].

F1 = {input, output, function}

Input = {q1, q2, q3, ….,qn} – set of query terms$1 \leq i \leq n$

Output = {f1, f2, f3,…., fm} – set of frequent terms$1 \leq i \leq m1 \leq i \leq n$

Function={I, sup, frequent terms, infrequent terms, cont,}

I-> {I1,I2,…}….set of query items in database

Sup->support count

Cont-> for each node counter sis maintained

FP tree construction:

Pass I: i. scan query log and find support

Sup=Pr (I1∪ I2)

ii. Discard infrequent items.

iii. Sort frequent items in decreasing order

Pass II: hash map->(Nodes = items, counter)

i. reads (T) at a time & map it to a path.

ii. Path can overlap when T share terms

cont++

hashmap ->(queries, count)

Sorting

sorted map->tree map(hash map)

Graphs generated from query log:

1. If two queries issued consecutively by many users occur frequently are good reformulations of each other.
QRG ={V_{QR},E_{QR}}
2. If two or more queries lead to click frequently on the same set of URLs

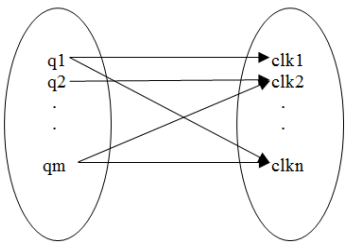

Fig.1 One to many relationship between query and click
QCG={VQC,EQC}

3. To use above both properties, we merge 1 and 2 into fusion graph
QFG ={V_{QF},E_{QF}}

Functions

F = {F1, F2}

F1=Function to calculate query relevance

Input = {QFG, g, d, numRWs , q}

Output={$Rel^F q$}

QFG-Query fusion graph

j-jump vector

d-damping factor

NumRWs-No of random walks

q-query keywords

RelFq-Fusion relevance vector for q

$rel^F q(q) = S[i]$

i-no of queries whose relevance value are approximately or equal to q then form a query group as S

S= {s1,s2,…sk}…1≤ i≤ k

F1- Function to find relevance between the user's latest singleton query group sc= {qc,clkc} and existing query groups si ∈ S

F1 = {input, output}

Input = {cxt_s}

cxt_s = Context vector of query group s

$cxt_s = w_{recency} \sum_{j=1}^{k} (1 - w_{recency})\, rel(q_{sk},clk_{sk})$

$w_{recency}$ = constant

Output = {Threshold integer value}

$Sim_{rel}(sc,si)=\sum_{q \in I(sc) \cap I(si)} rel(q_c,clk_c)(q)$
$*\sum_{q \in I(sc) \cap I(si)} cxt_{si}(q)$

$Sim\ (s_c,s_i) > T_{sim}$ $0 \leq T_{sim} \leq T_{max}$

T_{max}=Highest similarity

T_{sim}=Threshold value of similarity

Qs_c= singleton query group

Qs_i = existing groups

I = Image of query group

q = current query

$Sim(s_c,s_i) > Tsim$ is true

S = si ∪ sc…Add to existing group

O/w

S = S ∪ sc. Form new group

Output = { q1, q2, q3…}..List of query suggestions to current query 1≤ k≤ n

4. **Success**: q= =S

{Query match with existing query group}.

Z = { q1 , q2 ,q3…} 1≤ k≤ n

5. **Failure:**

i. Query did not match with existing cluster we can't display query suggestions**.**

ii. query entered by user is irrelevant keyword e.g. *azgkjllll*

Z= ∅

4. A Proposed System

While performing online complex tasks, user wants same information which is already searched by him; but at that moment user forgets the exact query. Sometimes user has seen specific document and even kept it but where..? These are the major problems faced by users ; so now a day's most of search engines provide facility of recording search history, bookmarking etc. But these facilities are not enough to satisfy diverse requirements of users. Some researchers have tried to use clustering approach for recording search history in systematic manner, but some clustering methods are quite expensive. Some are time based, text based that does not form efficient query groups.

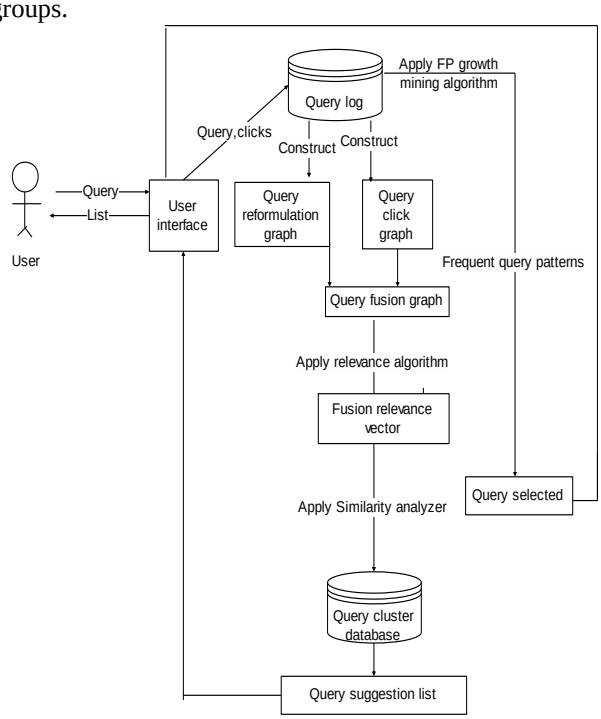

Fig.2 System architecture

In this proposed system, dynamic clustering is done over query fusion graph that combines probabilistic query reformulation graph, which captures relationship between queries frequently issued together by user, and a query click graph which captures relationship between queries frequently leading to clicks on similar urls. And then

combining the query reformulation graph and query click graph into single graph called query fusion graph; calculating query relevance over query fusion graph.

In this way, first approach mines frequent query patterns from users search history using FP growth algorithm, if user wants to select any query from his//her previous search log then he /she can. Otherwise he will enter new query and second approach identifies clusters of similar queries from all users search log the cluster of queries which is similar to current query are query suggestions.

System architecture

The figure 2 shown above describes the flow of the system. User logins to the system then frequent query patterns from only his/her search history are displayed then if user want to search data related to his search history then he/she will select query from frequent query patterns; otherwise select new query. The system takes input as I= $\{q_c, clk_c\}$ current users query q and corresponding clicks. In this system for query clustering purpose we are maintaining user search log. The search history of many of users contains metrics about query relevance, such as queries issued closely together are good reformulations of each other and queries share similar URLs; in this way using search behavioral data reformulation graph and click graph is constructed and later both graphs combined into query fusion graph. Now we will apply query relevance algorithm which takes input as query fusion graph, jump vector, number. of random walks and given user query q. This algorithm gives fusion relevance vector for q $rel^F q$, according to that will decide where to merge current query in query fusion graph. In this system to represent the relevance of other queries to this query image concept used. For each query group, we maintain a context vector, which combines the images of its member queries to form an overall representation. to form query groups similarity function sim_{rel} for two query groups based on the concepts of context vectors and query images. So this approach identifies clusters of similar queries from all users search history clusters of queries which is similar to current query are query suggestions.

5. Experimental Results

We obtained query reformulation graphs and query click graphs by merging a number of search logs. Each snapshot of the query log adds approximately new nodes and edges in the graph compared to the exactly preceding snapshot, to reduce the effect of noise and outliers; we maintain the query reformulation graph by keeping only query pairs that appeared at least two times,

and the query click graph by keeping only query click edges that had at least 3 clicks.

The particulars about platform and technology used:

- Base Operating System: Windows 7
- Databases: My SQL
- Web Server: Apache
- Language: Java
- Browser: IE8 & above, Mozilla Firefox, Google Chrome, Opera etc

As mentioned in mathematical model relevance between the user's latest singleton query group and existing query groups is calculated, that relevance value vary according to clusters as shown in fig.3 x-axis represents Groupid of each cluster and y-axis represents relevance value calculated between the user's latest singleton query group and existing query groups.

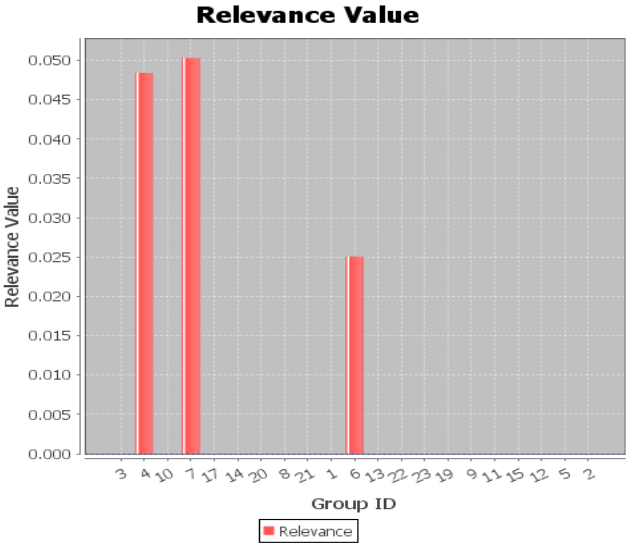

Fig.3 Varying relevance value of a query according to cluster

Either reformulation is occur or clicks are shared with any existing cluster then query is having highest relevance value, if partially that conditions are satisfied then having average relevance value. If both are not satisfied then having very less or 0 relevance value.

Query suggestion results

In table1 the query suggestion results of this system (QFG+ mining) and those from Google, Yahoo!, Live Search, and AOL are displayed. The query suggestions generated by this system are generally as good as those from commercial search engines. For some queries, this system gives even better query suggestion.

Conclusion

In this paper a new method for query suggestion system to satisfy the internet users for finding the information on web is presented. If user want to search same information again then no need to recall the query; our system will fulfill their requirement by giving better query suggestion which is already recorded in search log. In this first approach mines frequent query patterns sand second approach identifies clusters of queries from all users search history. In the proposed system resultant output is query suggestion list.Integration of both approaches increases precision of results of users.

References

M. Ester, H.-P. Kriegel, J. Sander,(1996) A density based algorithm for discovering clusters in large spatial databases with noise in *KDD*, pp. 226–231.

D.Beeferman and A. Berger,(2000) Agglomerative clustering of a search engine query log *In Proceedings of the sixth ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, pages 407 – 416.

Lyes Limam ,David Coquil, Harald Kosch ,Lionel Brunie,(2000) Extracting user interests from search log: A clustering approach, *Proceeding of the 2010 Workshops on Database and Expert Systems Applications*, Pages 5-9

Ji-Rong Wen, Jian-Yun Nie, (2001) Query Clustering Using Content Words and User Feedback , *ACM conf . on research and development in information retrieval* , Pages 442-443

J.Wen, J.Nie, and H.Zhang , (2001) Clustering user queries of a search engine ,*In Proceedings of the Tenth International World Wide Web Conference, Hong-Kong, China*, 1-5, pages 162–168

J.Wen, J.Nie, and H.Zhang.(2002) Query clustering using user logs *ACM Transactions on Information Systems*, 59–81

Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao,(2004) Mining frequent patterns without candidate generation : A Frequent Pattern tree approach, *Data Mining and knowledge discovery* ,8,53-87

R. Baeza-Yates, C. Hurtado, and M. Mendoza(2004), Query recommendation using query logs in search engines, *in EDBT*

Florian Verhein(2008) An Introduction to Frequent Pattern Growth (FP Growth) Algorithm.

Harksoo Kim,Jungyun Seo,(2008), Cluster-based FAQ retrieval using Latent term weights,*Journal of Natural Language Processing* , *IEEE* 15 41-1672/08

P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna,(2008),The Query-Flow Graph: Model and Applications, *Proc. 17thACM Conf. Information and Knowledge Management (CIKM*),

Toru Onoda, Takayuki Yumoto,(2008) Extracting and Clustering Related Keywords based on History of Query Frequency, *2008 Second International Symposium on Universal Communication*

J. Yi and F. Maghoul,(2009), Query Clustering Using Click-through Graph,*Proc. the 18th Int'l Conf. World Wide Web (WWW '09)*

Yuan Hong, Jaideep, Vaidya and Haibing Lu,(2011), Search engine query clustering using Top-K Search Results, *IEEE/WIC/ACM International Conferences on Web intelligence and intelligent Agent technology*,DOI10.1109/WI-IAT.2011.224

Hwang, H. W. Lauw, L. Getoor, and A. Ntoulas.(2012) Organizing user search histories,*IEEE Transactions on Knowledge and Data Engineering*, vol.,no 24(5):912–925

Kajal Y.VYAS,(2012),Improved web search result rank optimization using search engine query log, *Journal of information knowledge and research in Computer Engineering* ISSN:0975-6760 2012 volume-02,Issue-0.