International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Searching Queries Using Feedback Sessions

Swati S. Chiliveri[1], Pratiksha C. Dhande[2]

[1]Pune University, G.H.R.I.E.T., City-Pune, India

[2] G.H.R.I.E.T, Pune University, City-Pune, India

Abstract: ***For a very large query, different users have different search goals when they submit it to a search engine. To improve search engine relevance and user experience, the inference and analysis of user search goals can be convenient, relevance and user experience. Here we provide an overview of the system architecture of proposed feedback session framework with their advantages. Also we have detail deliberate the literature survey. First, we propose a framework based on clustering the proposed feedback sessions to detect different user search goals for a query. Using user click -through logs Feedback sessions are constructed and these sessions can efficiently show the needed information for user. Information needs of users. Second, we propose a novel approach to generate pseudo-documents for better representation of the feedback sessions for clustering second, we propose a novel approach to generate pseudo-documents to better represent the feedback sessions for clustering. Finally, to evaluate the performance of inferring user search goals we propose a new criterion "Classified Average Precision (CAP)".***

Keywords: Classified Average Precision (CAP), Feedback Sessions, Open Directory Project (ODP), Pseudo-documents, Restructuring search results, User search goals.

1. Introduction

In web search applications, user submits the queries to search engines to represent the information needs. However, sometimes requests that are submitted to search engine may not accurately represent user's specific information needs. Since many confusing queries may cover a wide topic and different users may want to get the different information on different aspects when they submit the same query.

Now-a-days, large amount of information is available on the Internet; Web search has become an indispensable tool for Web users to gain desired information. But, it becomes very hard task to get exact information that user want. Typically, Web users submit a short Web query consisting of a few words to search engines. Because these queries are short and ambiguous, how to interpret the queries in terms of a set of target categories has become a major research issue.

For example, when the query "apple" is submitted to a search engine, some people want to locate the natural fruit, while some people want to learn the different types of smart phones. Therefore, it is necessary to capture different user requirements, user needs in information retrieval. We define user search goals as the information on different aspects of a query that user groups want to obtain. Information need is a user's particular desire to obtain information to satisfy different user needs. Here, User search goals can be considered as the clusters of information needs for a query that has been submitted to search engine.

User search goals or query intent can have a lot of advantages in order to improve the search engine relevance and user experience. Here, some advantages of using this system are:

1) It is possible to restructure web search results according to user search goals by grouping the search results with the same search goal. Users with different search goals can easily find what they want and satisfy the users need.
2) User search goals that are represented by some keywords can be utilized in query recommendation thus, the suggested queries can help users to form their queries more precisely and with more accurately.
3) The distributions of user search goals can also be used in applications such as re-ranking web search results that contain different user search goals.

2. Literature Survey

Other works analyze the search results returned by the search engine directly to exploit different query aspects. However, query aspects without user feedback have some drawbacks to improve search engine. Some works take user feedback into account and analyze the different clicked URLs of a query in user click-through logs directly, but the number of different clicked URLs of a query may not be large enough to get ideal results. The early utilization of user click-through logs is to obtain user implicit feedback to enlarge training data when learning ranking functions in information retrieval. Here we are studying how to infer user search goals from user click-through logs and restructure the search results according to the inferred user search goals.

2.1 Automatic Identification of User Goals inWeb Search

By Uichin Lee, Zhenyu Liu, Junghoo Cho [12]
In this paper, he proposed two types of features for the goal-identification task:
1. Anchor-Link Distribution
2. User-Click Behavior

1. Anchor-Link Distribution
For a given a query, its anchor-link distribution is determined as follows: Firstly, find all the anchors appearing on the Web that have the identical text as the query, and extract their destination URLs. Then, count the numbers of times each destination URL appears in this list .After getting the count of destination URL; arrange the destinations in the descending order of their appearance. Then writer created a

histogram where the frequency count in the bin is the number of times that the ith destination appears. Finally, normalize the frequency in each bin so that all frequency values add up to 1.

2. User-Click Behavior

A. Click Distribution

In this paper [12], he proposed that the users goal for a given query can be learned from how users in the past have communicated with the returned results for the particular query. If the goal of a query is navigational, then in the past users should have mostly clicked on a single Website corresponding to the one they have in his mind.

B. Average Number of Clicks per Query.

Besides click distribution, they need to be attention on another feature embedded in the user-click behavior is how many results a user clicks on when the query is issued. Generally, for a navigational query, the user is most likely to click on only one result that corresponds to the Website the user has in mind.

2.2 Learn from Web Search Logs to Organize Search Results

By Xuanhui Wang ChengXiang Zhai[6]

For a given input query to the search engine, the general procedure of this proposed approach is described as follows [6]:

1) Get its connected information from search engine logs. All the detail forms a working set.
2) Study the aspects from the information in the working set. These aspects correspond to user's regards given the input query. Each aspect is labeled with a corresponding query.
3) Classify and arrange the search results of the input query according to the aspects studied above.

1. Locating Related Past Queries

Given a query k, a search engine will return a ranked list of Web pages. To know what the users are really interested in given this query, author first retrieves its past similar queries from preprocessed history data collection. Consider that there are N pseudo-documents in history data set: H = {K_1, K_2, ..., K_N}. Each K_i corresponds to a unique query and is enriched with click-through information. To find k's related queries in H, a natural way is to use a text retrieval algorithm. In this paper, author have used the OKAPI method, it is one of the state-of-the art retrieval method. After calculating the similarity between query k and pseudo-document K_i , based on the similarity scores, they rank all the documents in H. The top ranked documents provide us a working set to learn the aspects that users are usually interested in. Each document in H related to a past query, and thus the top ranked documents related to k's corresponding past queries.

2. Studying the Aspects by Clustering

Given a query k, H_k = {e_1, ..., e_n} represent the set of top ranked pseudo-documents from the past collection H. These pseudo-documents contain the aspects that users are interested in. In order discover the learning aspect we need to use the clustering method. Any clustering algorithm could be applied here like K Means, C Means Clustering algorithm, etc. In this work, author has used an algorithm based on graph partition called the star clustering algorithm. A good property of the star clustering is that it can suggest a good label for each cluster naturally. It outputs a center for each cluster.

3. Differentiating the Search Results

In order to organize the search results according to users interests, we need to use the learned aspects from the related past queries to categorize the search results. Given the top m Web pages returned by a search engine for k: q_1, …., q_m , group them into different aspects using a categorization algorithm. In this paper, author has used a simple centroid-based technique for differentiated.

2.3 Building Bridges for Web Query Classification

By Dou Shen, Jian-Tao Sun, Qiang Yang, Zheng Chen [10]

In this paper , author present a novel method for query classification that perform better than the winning solution of the ACM KDDCUP 2005 competition, whose main scope is to classify 800,000 real user queries. He, first build a bridging classifier on an intermediate taxonomy in an offline mode. This classifier is then used in an online mode to map user queries to the target group via the above intermediate taxonomy. There are three Classification Approaches, they are as follows

a) Classification Approaches

A. Classification by Exact Matching:

In this approach, exact matching technique produce classification results with high precision but low recall. Exact matching technique produces high precision because this method depends on the Web pages which are associated with the manually annotated category details. It produces low recall because many search result pages have no intermediate classification. The exact matching method cannot find all the mappings from the existing intermediate taxonomy to the target taxonomy which also concludes in low recall.

B. Classification by SVM

In this approach, Support Vector Machine (SVM) was used as a base classifier.

C. Classifiers by Bridges

In this approach, author describes new query classification method called taxonomy bridging classifier, or bridging classifier. It provides the relationship between the target taxonomy and queries by taking an intermediate taxonomy as a bridge.

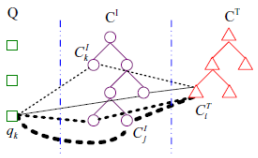

Figure 1: Taxonomy Bridging Classifier

2.4 Grouper: A Dynamic Clustering Interface To Web Search Results

By O. Zamir and O. Etzioni[13]

In this paper, author used Suffix Tree Clustering (STC) to recognize set of documents having common phrases and then create cluster based on these phrases or contents. In this approach, author used documents pieces instead entire document for clustering web documents. However, generating meaningful labels for clusters is one of the most demanding tasks in document clustering. So, to overcome this problem faced, author used a supervised learning method to extract possible phrases from search result pieces or contents and these phrases are then used to cluster web search results.

Suffix Tree Clustering (STC)
Suffix Tree Clustering is an incremental, linear time (in the document collection size) algorithm, which creates clusters based on phrases shared between documents. It satisfies the rigorous requirements of the Web domain. It is shown that STC is faster than quality of clustering methods in this domain, and also prove that Web document clustering via STC is both feasible and potentially beneficial. STC does not treat a document as a set of words but rather as a string, making use of closest information between words. Suffix Tree Clustering mainly relies on a suffix tree to recognize sets of documents that share common phrases and uses this information to create clusters and conclude their contents for users successfully. In this way Suffix Tree Clustering is used to search web search results.

3. System Implementation

3.1 System Architecture

Fig. 2 shows the System architecture of Inferring User Search Goals with Feedback Sessions. This proposed framework consists of two parts divided by the dashed line.

In the upper part, all the feedback sessions of a query are first extracted from user click-through logs and mapped to pseudo-documents. Then, user search goals are inferred by clustering these pseudo-documents and depicted with some keywords.

Initially, we do not know the exact number of user search goals in advance. So, author have tried several different values and determined the optimal value by the feedback from the bottom part.

In the bottom part, the original search results are restructured based on the user search goals inferred from the upper part. Then, author evaluates the performance of restructuring search results with the help of evaluation criterion CAP. And the evaluation result will be used as the feedback to select the optimal number of user search goals in the upper part.

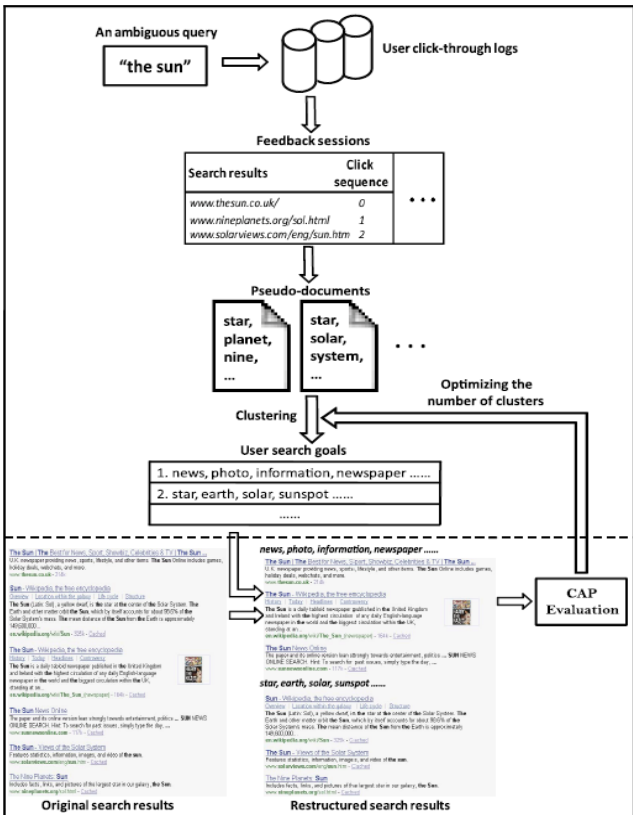

Figure 2: Inferring User Search Goals with Feedback Sessions

3.2 Mathematical Model

Here, we propose a novel way to map feedback sessions to pseudo-documents, as illustrated in Fig. 3. The building of a pseudo-document includes two steps. They are described in the following:

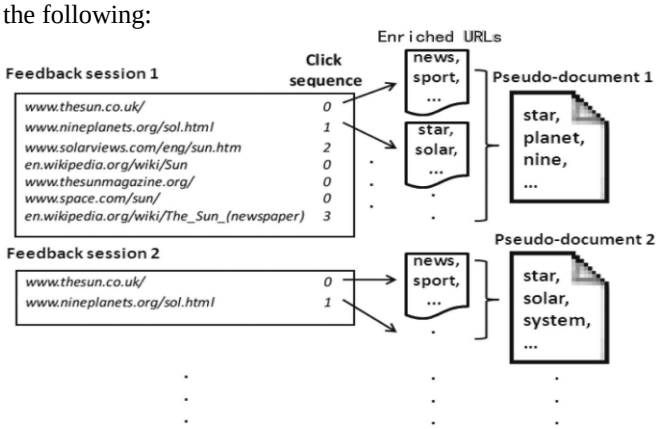

Figure 3: Illustration for mapping feedback sessions to pseudo-documents.

A. Representing the URLs in the feedback session

Each URL in a feedback session is represented by a small text paragraph that consists of its title and its piece.
Each URLs title and piece are represented by a Term Frequency-Inverse Document Frequency (TF-IDF) vector [14], respectively, as in

$$T_{ui} = [t_{w1}, t_{w2}, \ldots, t_{wn}]^T$$
$$S_{ui} = [s_{w1}, s_{w2}, \ldots, s_{wn}]^T \quad (1)$$

Where T_{ui} and S_{ui} are the TF-IDF vectors of the URLs title and snippet, respectively. u_i means the i^{th} URL in the

feedback session. And w_j (j=1,2,….,n) is the jth term appearing in the enriched URLs. We represent the enriched URL by the weighted sum of Tui and Sui, namely

$$F_{ui}=w_tT_{ui}+W_sS_{ui}= [f_{w1}, f_{w2},…., f_{wn}]^T \quad (2)$$

Where F_{ui} means the feature representation of the i^{th} URL in the feedback session, and w_t and w_s are the weights of the titles and the pieces, respectively. In this way, we represent the URLs in the feedback session.

B. Forming pseudo-document based on URL representations

In order to obtain the URL representation of a feedback session, we propose an optimization technique to combine both clicked and unclicked URLs in the feedback session.
Let F_{fs} be the URL representation of a feedback session, and $f_{fs}(w)$ be the value for the term w.

Let F_{ucm} (m =1, 2,.....,M) and F_{ucl} (l=1, 2, . . . , L) be the URL representations of the clicked and unclicked URLs in this feedback session, respectively. Let $f_{ucm}(w)$ and f_{ucl} (w) be the values for the term w in the vectors.

$$F_{fs}=\left[f_{fs}(w_1), f_{fs}(w_2),... f_{fs}(w_n)\right]^T,$$

$$f_{fs}(w)=\arg\min_{f_{fs}(w)}\left\{\sum_M\left[f_{fs}(w)-f_{uc_m}(w)\right]^2-\lambda\sum_L\left[f_{fs}(w)-f_{u\bar{c}_l}(w)\right]^2\right\}, f_{fs}(w)\in I_{c.} \quad (3)$$

Let I_c be the interval [μf_{uc} (w) - σf_{uc} (w), μf_{uc}(w) + σf_{uc} (w)] and $I_{\bar{c}}$ be the interval [$\mu f_{u\bar{c}}$ (w) - $\sigma f_{u\bar{c}}$ (w), $\mu f_{u\bar{c}}$ (w) + $\sigma f_{u\bar{c}}$ (w)], where μf_{uc}(w) and σf_{uc}(W) represent the mean and mean square error of f_u(w) respectively, and $\mu f_{u\bar{c}}$ (w) and $\sigma f_{u\bar{c}}$ (w) represent the mean and mean square error of $f_{u\bar{c}}$ (w), respectively.

$$f_{fs}(w) = 0, I_c \subseteq I_{\bar{c}} \subseteq Ic_{\bar{c}} \subseteq I_c \quad (4)$$

λ is a parameter balancing the importance of clicked and unclicked URLs. In this way, we form the pseudo-documents based on URL Representations.

C. Inferring user search goals by clustering pseudo-documents

With the proposed pseudo-documents, we can search goals for inferred user.
As in (3) and (4), each feedback session is related by a pseudo-document and the feature representation of the pseudo-document is F_{fs}. The similarity between two pseudo-documents is computed as the cosine score of F_{fsi} and F_{fsj} , as follows:

$$Sim_{i,j}=\cos\left(F_{fs_i},F_{fs_j}\right)$$
$$=\frac{F_{fs_i}.F_{fs_j}}{|F_{fs_i}||F_{fs_j}|}. \quad (5)$$

And the distance between two feedback sessions is

$$Dis_{i,j} = 1 – Sim_{i,j} \quad (6)$$

We cluster pseudo-documents by Fuzzy C-means clustering which is simple and effective.

After clustering all the pseudo-documents, each cluster can be considered as one user search goal. The center point of a cluster is computed as the average of the vectors of all the pseudo-documents in the cluster, as shown in

$$F_{center_i}=\frac{\sum_{k=1}^{C_i}F_{fs_k}}{C_i},\left(F_{fs_k}\subset Clusteri\right), \quad (7)$$

Where F_{center_i} is the i^{th} clusters center and C_i is the number of the pseudo-documents in the i^{th} cluster. F_{center_i} is utilized to conclude the search goal of the i^{th} cluster. Hence, we infer the user search goals by clustering the pseudo-documents.

D. Evaluation Based on Restructuring Web Search Results

1. Restructuring Web Search Results

The inferred user search goals are represented by the vectors in (7) and the feature representation of each URL in the search results can be computed by (1) and (2). From equation (1) and (2) we can compute the feature representation of URL. Then, we can categorize each URL into a cluster centered by the inferred search goals. Like this we restructure the Web Search Results.

2. Evaluation Criterion

Because from user click-through logs, we can get implicit relevance feedbacks, namely “clicked” means relevant and “unclicked” means irrelevant. As we mainly focus on clicked and unclicked URLs, here we use the average precision to calculate the performance of restructured results. A possible evaluation criterion is the average precision (AP) [14] which evaluates according to user implicit feedbacks.

$$AP=\frac{1}{N^+}\sum_{r=1}^{N}rel(r)\frac{R_r}{r}, \quad (8)$$

Where N^+ is the number of relevant (or clicked) documents in the retrieved ones, r is the rank, N is the total number of retrieved documents, rel() is a binary function on the relevance of a given rank, and R_r is the number of relevant retrieved documents of rank r or less.

We first introduce “Voted AP (VAP)” which is the AP of the class including more clicks namely votes.

We propose the risk as follows:

$$Risk=\frac{\sum_{i,j=1(i<j)}^{m}d_{i,j}}{C_m^2} \quad (9)$$

It calculates the normalized number of clicked URL pairs which are not present in the same class, where m is the number of the clicked URLs. We can more extend VAP by establishing the above Risk and propose a new criterion "Classified AP," as shown below

$$CAP = VAP \times (1 - Risk)^{\gamma} \quad (10)$$

From (10), we can see that CAP selects the AP of the class that user is interested in (i.e., with the most clicks/votes) and takes the risk of wrong categorization into account. And γ is used to adjust the influence of Risk on CAP, which can be learned from training data. Finally, in this way, we evaluate the performance of restructuring web search results using CAP.

4. Algorithm

Fuzzy C-Means Clustering

I. Initialize U= $[u_{ij}]$ matrix, $U^{(0)}$

II. At k-step: calculate the centers vectors $C^{(k)}=[c_j]$ with $U^{(k)}$

$$c_j = \frac{\sum_{i-1}^{N} u_{ij}^{m} . x_i}{\sum_{i-1}^{N} u_{ij}^{m}}$$

III. Update $U^{(k)}$, $U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{k-1}^{c} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}}$$

IV. If $\| U^{(k+1)} - U^{(k)} \| < \xi$ then STOP; otherwise return to step 2.

5. Discussions and Results

From Fig. 4, we can see that the Admin Logs in to the account.

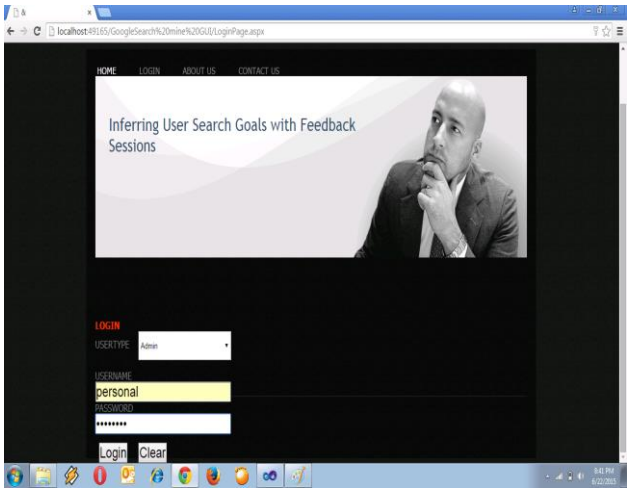

Figure 4: Admin Login Page

We can do the registration through Admin Login Page as shown in Fig. 5 and we also provide the necessary validations on the text boxes.

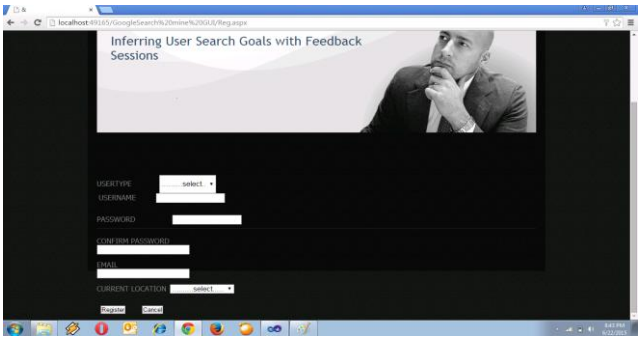

Figure 5: Registration Page

First the user creates the account through registration and then Logs in to the account as shown in fig. 6.

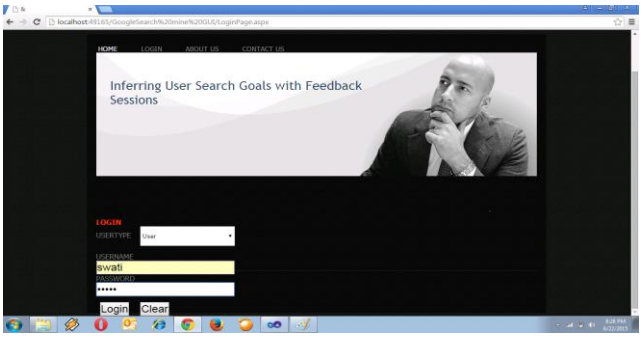

Figure 6: Login Page

After logging in, the user can do the customize search on clicking the radio button as shown in fig. 7.

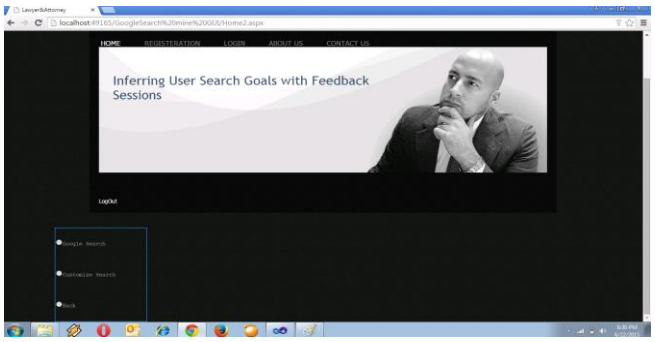

Figure 7: Home page

The user searches for the required query which is shown in fig. 8

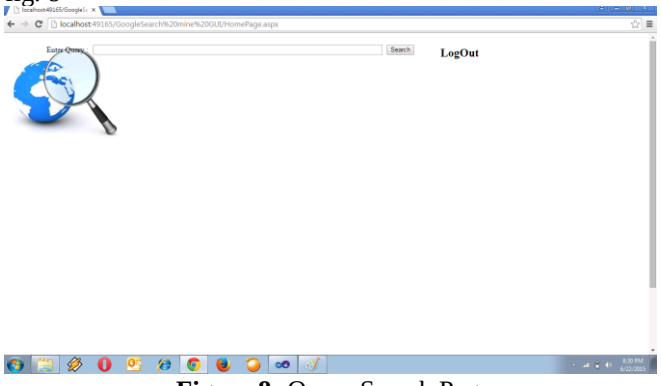

Figure 8: Query Search Page

Firstly, we get the feedback sessions to be analyzed to infer user search goals rather than search results or clicked URLs. The clicked URLs and the un-clicked ones both before the last click are considered as user implicit feedbacks and taken into account to construct a feedback session which is shown in fig. 9.

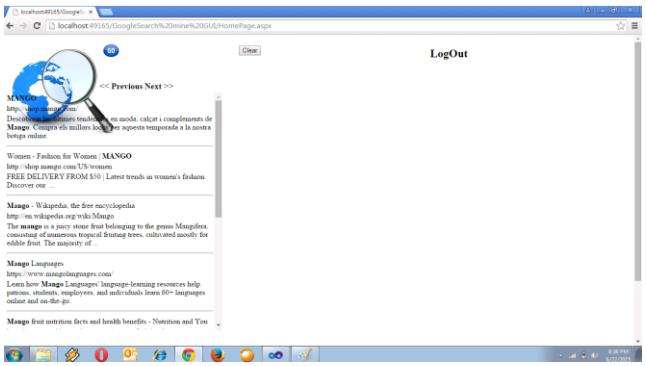

Figure 9: Searched query

Therefore, feedback sessions give the user information needs more efficiently.
Secondly, we map feedback sessions to pseudo-documents to approximate goal texts in user minds which is shown in Fig.10

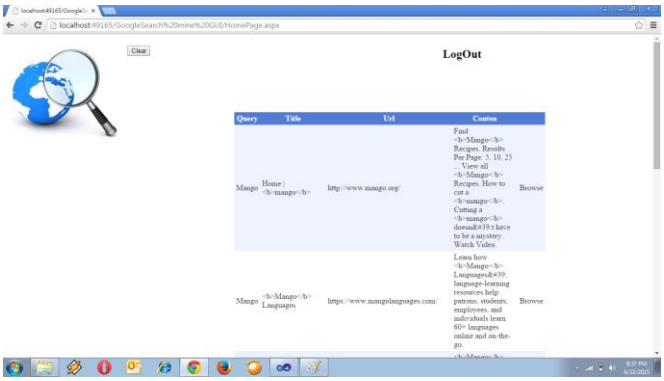

Figure 10: Feedback Session Page

At last, we cluster these pseudo documents to infer user search goals and depict them with some keywords to get the restructured web search results which is shown in fig.11.

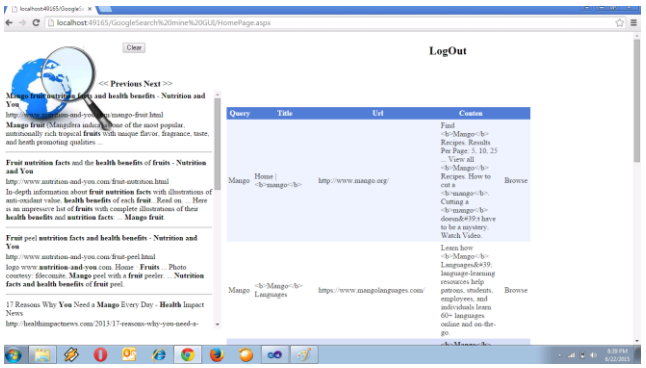

Figure 11: Restructured Web Search Result

6. Conclusion

Hence, we have concluded that a novel approach has been proposed to infer user search goals for a query by Fuzzy C Means clustering to get its feedback sessions represented by pseudo-documents. Firstly, we get the feedback sessions to be analyzed to infer user search goals rather than search results or clicked URLs. The clicked URLs and the un-clicked ones both before the last click are considered as user implicit feedbacks and taken into account to construct feedback sessions. Therefore, feedback sessions gives the user information needs more efficiently. Secondly, we map feedback sessions to pseudo-documents to approximate goal texts in user minds. Based on these pseudo-documents, user search goals can then be discovered and depicted with some keywords. Finally, a new criterion known as CAP is formulated to evaluate the performance of user search goal inference.

References

[1] A. Bonnaccorsi, “On the Relationship between Firm Size and Export Intensity,” Journal of International Zheng Lu, Student Member, IEEE, Hengyang Zha, Xiaokang Yang, Senior Member, IEEE, Weiyao Lin, Member, IEEE, and Zhaohui Zheng -A New Algorithm for Inferring User Search Goals with Feedback Sessions- IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 3, MARCH 2013. (journal style)

[2] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, Context Aware Query Suggestion by Mining Click-Through, Proc. 14th ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining (SIGKDD 08), pp. 875-883, 2008.

[3] R. Jones and K.L. Klinkner,Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs, Proc. 17th ACM Conf. Information and Knowledge Management (CIKM 08), pp. 699-708, 2008.

[4] X. Li, Y.-YWang, and A. Acero, Learning Query Intent from Regularized Click Graphs, Proc. 31st Ann. Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 08), pp. 339-346, 2008.

[5] B. Poblete and B.-Y Ricardo, Query-Sets: Using Implicit Feedback and Query Patterns to Organize Web Documents, Proc. 17th Intl Conf. World Wide Web (WWW 08), pp. 41-50, 2008.

[6] X.Wang and C.-X Zhai,Learn from Web Search Logs to Organize Search Results, Proc. 30th Ann. Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 07), pp. 87-94, 2007.

[7] S. Beitzel, E. Jensen, A. Chowdhury, and O. Frieder, Varying Approaches to Topical Web Query Classification, Proc. 30th Ann. Intl ACM SIGIR Conf. Research and Development (SIGIR 07), pp. 783 784, 2007.

[8]] M. Pasca and B.-V Durme, What You Seek Is what You Get: Extraction of Class Attributes from Query Logs, Proc. 20th Intl Joint Conf. Artificial Intelligence (IJCAI 07), pp. 2832-2837, 2007.

[9] R. Jones, B. Rey, O. Madani, and W. Greiner, Generating Query Substitutions, Proc. 15th Intl Conf. World Wide Web (WWW 06), pp. 387-396, 2006.

[10] D. Shen, J. Sun, Q. Yang, and Z. Chen, Building Bridges for Web Query Classification, Proc. 29th Ann.

Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 06), pp. 131-138, 2006.

[11] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay, Accurately Interpreting Clickthrough Data as Implicit Feedback, Proc. 28th Ann. Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 05), pp. 154-161, 2005.

[12] . Lee, Z. Liu, and J. Cho, Automatic Identification of User Goals in Web Search, Proc. 14th Intl Conf. World Wide Web (WWW 05), pp. 391-400, 2005.

[13] Zamir, O. And Etzioni, O. 1999. Grouper: A dynamic clustering interface to Web search results. Comput. Netw.31, 1116, 13611374.

[14] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval ACM Press, 1999.