# A Metric for Inferring User Search Goals in Search Engines

**M. Monika, N. Rajesh, K.Rameshbabu**

***Abstract*— For a broad topic, different users may have different search queries while they submit it to Search engines. In improving search engine relevance and user experience, the inference and analysis of user search goals can be very useful. In this paper, First we propose a way to infer user search goals by analysing the Feedback Session which is created from Search engine Query logs. Second we propose a new metric in generating Pseudo documents to better represent the Feedback Sessions for clustering. Finally we propose a new criterion to evaluate the performance of inferring user search goals.**

***Index Terms*— User search Queries, Feedback sessions, pseudo-documents, Ranking, restructuring search results.**

## I. INTRODUCTION

In online search Applications, different User Search queries are submitted to search engines to represent the information needs of users. Though, sometimes user search queries may not precisely represent users' specific information needs and different users may want to get different information when they submit the same query. For example, when the query "the sun" is submitted to a search engine, some users want to locate the homepage of a United Kingdom newspaper, while some others want to learn the natural knowledge of the sun, as shown in Fig. 1. Therefore, it is essential to capture different user search goals in retrieving information .We define user search goals as the information on different aspects of a user search query that user groups want to find. Information is a user's particular desire to satisfy his/her need. User search goals are considered as the clusters of information needs for a search query.

The inference and exploration of user search goals can have a lot of advantages in improving user experience and search engine relevance.

In this paper, our objective is to discover the diverse user search goals for a query and showing each goal with some keywords automatically. We propose a new approach to infer user search goals for a query by clustering our feedback sessions. The feedback Session is well-defined as the sequences of both clicked and unclicked search URLs along with their ranks and ends with the last URL that was clicked in a session from user search logs. Then, we propose a new optimization method to map feedback sessions with pseudo-documents which can efficiently return user information needs. At last, we Cluster these pseudo documents to infer user search goals and show them with some keywords.


Fig.1 The examples of the different user search goals and their distributions for the query "cat" by our experiment

Finally, we propose an evaluation criterion classified average precision (CAP) to evaluate the performance of the restructured web search results

## II. METRIC FOR INFERRING USER SEARCH GOALS

### A. *Framework of our approach*

Fig. 2 shows the framework of our approach by taking a example of the ambiguous Query "The Sun". It contains two parts divided by the dashed line.

In the upper part, all the feedback sessions of a user search query are first extracted from user search logs and mapped to Pseudo-documents. Then, user search goals are inferred by clustering these pseudo-documents and depicted with some keywords. In the bottom part, the original search results are restructured based on the user search goals inferred from the upper part. Then, we evaluate the performance of restructuring search results by our proposed 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.

Manuscript received September 02, 2014.
**M. Monika**,Department of computer science, Mallareddy engineering college for women,, Bahadurpally, Rangareddy Dist (TS), India
**N.Rajesh**,,Department of computer science, Mallareddy engineering college for women,, Bahadurpally, Rangareddy Dist (TS), India
**K.Rameshbabu,** Department of computer science, Mallareddy engineering college for women,, Bahadurpally, Rangareddy Dist (TS), India.

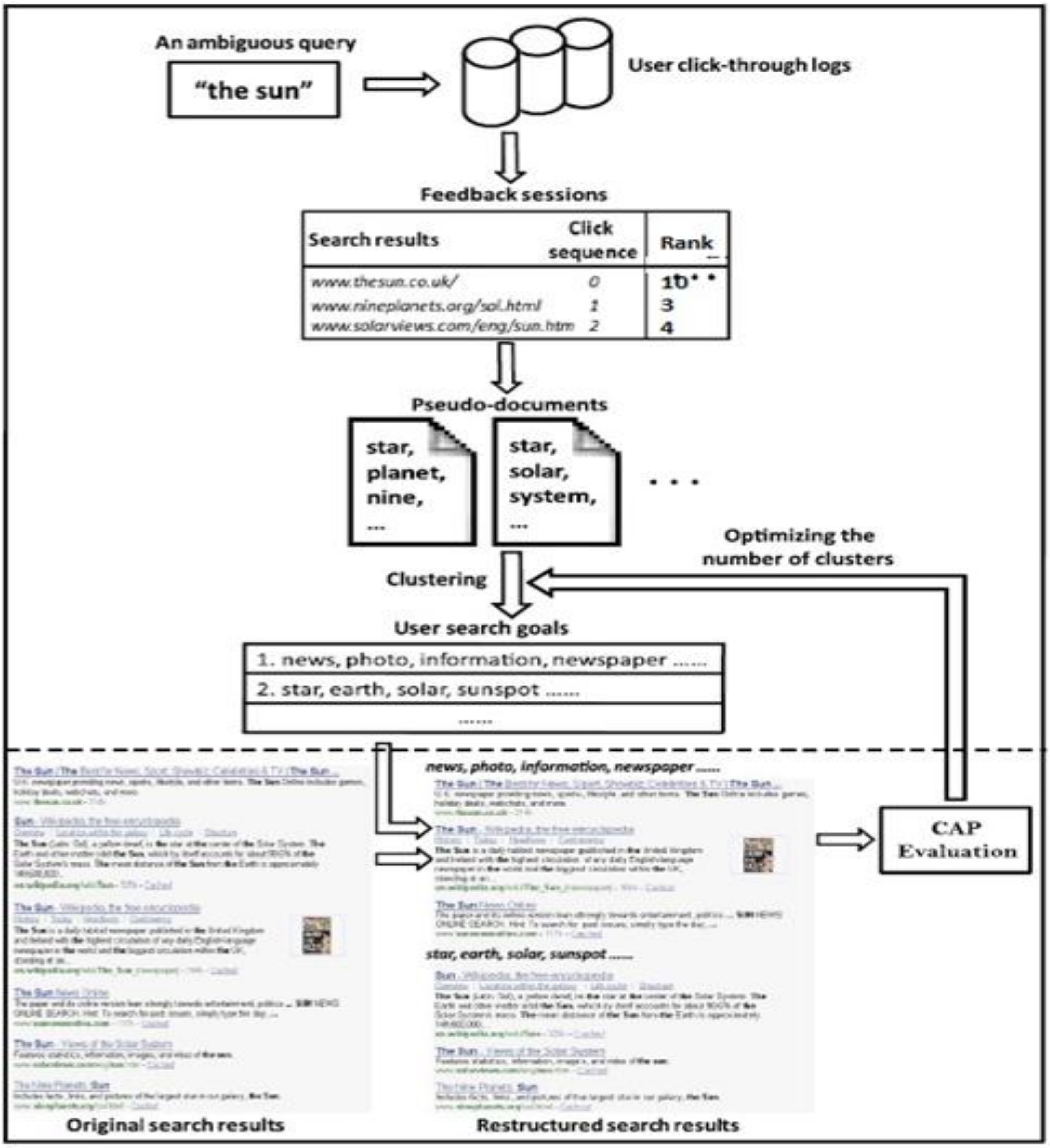



**Fig 2: Framework of Our Approach**

## III. Representation of Feedback Sessions

In this section, we first describe the proposed feedback sessions and then we present the proposed pseudo documents to represent feedback sessions.

In this paper, we focus on inferring user search goals for a particular search query. Therefore, the single session containing only one query is introduced, which differentiates from the conventional session. Meanwhile, the Feedback session in this paper is based on a single session, though it can be extended to the whole session.

The proposed feedback session consists of clicked URLS, unclicked URLs and web page rank of the URLS, ends with the last URL that was clicked in a single session. It is motivated that before the last click, all the URLs have been seen over and evaluated by users. Therefore, besides the clicked URLs, the unclicked ones before the last click and their corresponding page ranks should be a part of the user feedbacks.

In Fig. 3, the left part lists 10 search results of the query “the sun” and the right part is a user’s click sequence and rank where “0” means “unclicked.” And the numbers 1,2,3 shows the order of clicked URLS.The single session includes all the 10 URLs in Fig. 3, while the feedback session only includes the seven URLs in the rectangular box. The seven URLs consist of three clicked URLs and four unclicked URLs in this example.

| Search results | PAGE RANK | Click sequence |
|---|---|---|
| www.thesun.co.uk/ | 10 | 0 |
| www.nineplanets.org/sol.html | 3 | 1 |
| www.solarviews.com/eng/sun.htm | 4 | 2 |
| en.wikipedia.org/wiki/Sun | 10 | 0 |
| www.thesunmagazine.org/ | 12 | 0 |
| www.space.com/sun/ | 15 | 0 |
| en.wikipedia.org/wiki/The_Sun_(newspaper) | 9 | 3 |

Fig. 3. A feedback session in a single session. "0" in click sequence means "unclicked." All the 10 URLs construct a single session. The URLs in the rectangular box construct a feedback session.

B. Map Feedback Sessions to Pseudo-Documents

In this paper, we propose a new way to map feed-back sessions to pseudo-documents as illustrated in Fig 6.The building of a pseudo-document includes two steps. They are described in the following: For example, Fig. 4 shows a popular binary vector method to represent a feedback session

| Search results | Rank | Click sequence | Binary vector |
|---|---|---|---|
| www.thesun.co.uk/ | 10 | 0 | 0 |
| www.nineplanets.org/sol.html | 3 | 1 | 1 |
| www.solarviews.com/eng/sun.htm | 4 | 2 | 1 |
| en.wikipedia.org/wiki/Sun | 10 | 0 | 0 |
| www.thesunmagazine.org/ | 12 | 0 | 0 |
| www.space.com/sun/ | 15 | 0 | 0 |
| en.wikipedia.org/wiki/The_Sun_(newspaper) | 9 | 3 | 1 |

Fig 4 The binary vector representation of a feedback session.

For a query, users will usually have some vague keywords representing their interests in their minds. They use these keywords to determine whether a document can satisfy their needs. We name these keywords "goal texts" as shown in Fig. 5.

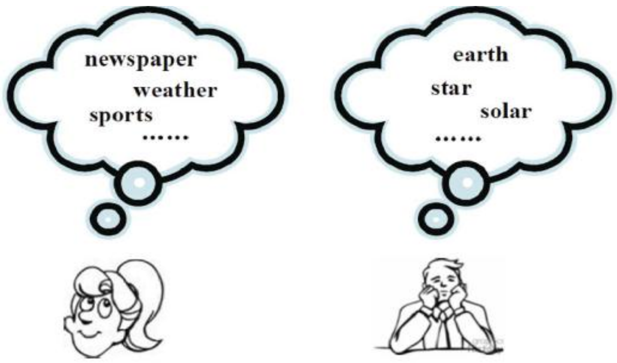



Fig.5 Goal texts. For a query, different users will have different keywords in their minds. These keywords are vague and have no order. We name them "goal texts," which reflect user information needs.

1) Representing the URLs in the feedback session

.

In the first step, we enrich the URLs with addition-al textual contents by extracting the titles and snip-pets of the returned URLs contained in the feedback session. Finally, each URL's title and snippet are represented by a Term Frequency-Inverse Document Frequency (TF-IDF) vector [1], respectively, as in

$$\mathbf{T}_{u_i} = [t_{w_1}, t_{w_2}, \ldots, t_{w_n}]^T, \quad \mathbf{S}_{u_i} = [s_{w_1}, s_{w_2}, \ldots, s_{w_n}]^T, \tag{1}$$

Where $T_{ui}$ and $S_{ui}$ are the TF-IDF vectors of the URL's title and snippet, respectively.$u_i$ means the ith URL in the feedback session. And $w_j$ (j=1,2,…n) is the jth term appearing in the enriched URLs. Here, a "term" is defined as a word or a number in the dictionary of document collections. $t_w$j and $s_{wj}$ represent the TF-IDF value of the $j_{th}$ term in the URL's title and snippet, respectively. Considering that URLs' titles and snippets have different significances, we represent the enriched URL by the weighted sum of $T_{ui}$ and $S_{ui}$ , namely

$$\mathbf{F}_{u_i} = \omega_t \mathbf{T}_{u_i} + \omega_s \mathbf{S}_{u_i} = [f_{w_1}, f_{w_2}, \ldots, f_{w_n}]^T, \tag{2}$$

where $F_{ui}$ means the feature representation of the ith URL in the feedback session, and $w_t$ and $w_s$ are the weights of the titles and the snippets, respectively Then, we stipulate that the titles should be more significant than the snippets. Therefore, the weight of the titles should be higher and we set $w_t$ to be 2 in this paper.

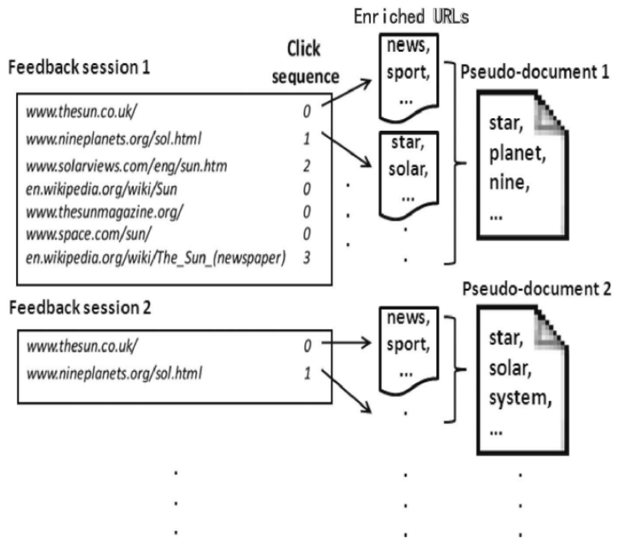



Fig 6: Illustration for mapping feedback sessions to pseudo-documents

2) Generate pseudo-document based on URL representations

In order to obtain the feature representation of a feedback session, we propose an optimization method to combine both clicked and unclicked URLs in the feedback session.

Let $F_{fs}$ be the feature representation of a feedback session, and $f_{fs(w)}$ be the value for the term w. Let $F_{ucm}$(m=1,2,…M) and $F_{uci}$ (m=1,2,….M) and $\mathbf{F}_{u\bar{c}_l}$ (l=1,2,…L) be the feature representations of the clicked and unclicked URLs in this feedback session, respectively. Let $f_{ucm(w)}$ and $f_{u\bar{c}_l}(w)$ be the values for the term w in the vectors. We want to obtain such a $F_{fs}$ that the sum of the distances between $F_{fs}$ and each $F_{ucm}$ is minimized and the sum of the distances between $F_{fs}$ and each $\mathbf{F}_{u\bar{c}_l}$ is maximized. Based on the assumption that the terms in the vectors are independent, we can perform optimization on each dimension independently, as shown in (3)

$$\begin{aligned} \mathbf{F}_{fs} &= [f_{fs}(w_1), f_{fs}(w_2), \ldots f_{fs}(w_n)]^T, \\ f_{fs}(w) &= \arg\min_{f_{fs}(w)} \Big\{ \sum_M [f_{fs}(w) - f_{uc_m}(w)]^2 \\ &\quad - \lambda \sum_L [f_{fs}(w) - f_{u\bar{c}_l}(w)]^2 \Big\}, f_{fs}(w) \in I_c. \end{aligned} \tag{3}$$

## IV. EVALUATION CRITERION

To apply the evaluation method to large set of data, the single sessions in user click-through logs are used to minimize manual work. Because from user click-through logs, we can get implicit relevance feedbacks, namely "clicked" means relevant and "unclicked" means irrelevant.

A possible evaluation criterion is the average precision (AP)[1] which evaluates according to user implicit feedbacks. AP is the average of precisions computed at the point of each relevant document in the ranked sequence, as shown in

$$AP = \frac{1}{N^+} \sum_{r=1}^{N} rel(r) \frac{R_r}{r}, \tag{4}$$

where $N^+$ is the number of 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 Rr is the number of relevant retrieved documents of rank r or less.

For example, Fig. 7a is a single session with user's implicit feedback and we can compute AP as: (1/4)X(1/2+2/3+3/7+4/9)=0.510. However, AP is not suitable for evaluating the restructured or clustered searching results. The proposed new criterion for evaluating restructured results is described in the following.

As shown in Fig. 7b, the URLs in the single session are restructured into two classes where the un-boldfaced ones in Fig. 3a are clustered into class 1 and boldfaced ones are clustered into class 2. We first introduce "Voted AP (VAP) which is the AP of the class including more clicks namely votes.

However, VAP is still an unsatisfactory criterion. Considering an extreme case, if each URL in the click session is categorized into one class, VAP will always be the highest value namely 1 no matter whether users have so many search goals or not. Therefore, there should be a risk to avoid classifying search results into too many classes by error.

We propose the risk as follows: Therefore, there should be a risk to avoid classifying search results into too many classes by error. We propose the risk as follows:

$$Risk = \frac{\sum_{i,j=1(i<j)}^{m} d_{ij}}{C_m^2}. \tag{5}$$

It calculates the normalized number of clicked URL pairs that are not in the same class, where m is the number of the clicked URLs. If the pair of the ith clicked URL and the jth clicked URL are not categorized into one class, $d_{ij}$ will be 1;otherwise, it will In the example of Fig. 7b, the lines connect the clicked URL pairs and the values of the line reflect whether the two URLs are in the same class or not. Then, the risk in Fig. 7b can be calculated by: Risk=3/6=1/2.Based on the above discussions, we can further extend VAP by introducing the above Risk and propose a new criterion "Classified AP," as shown below

$$CAP = VAP \times (1 - Risk)^{\gamma}. \tag{6}$$

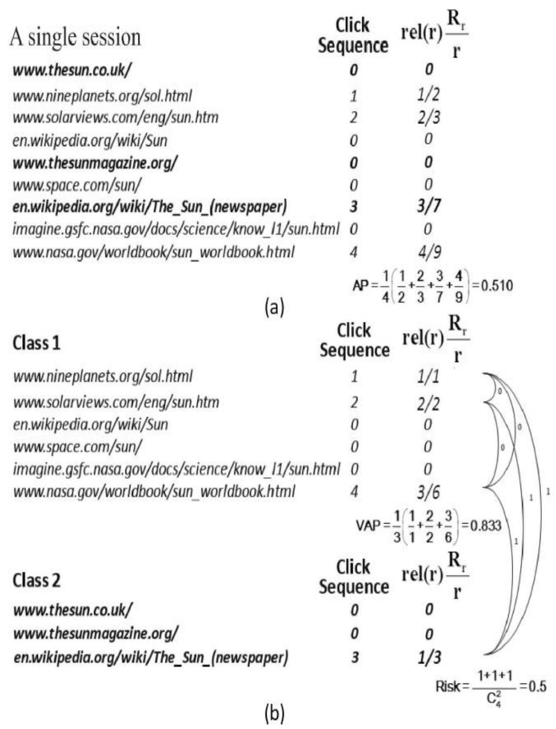



Fig 7 : Illustration for the calculation of AP, VAP, and Risk.

## V. EXPERIMENTS

In this section, we will present experiments of our proposed algorithm. The data set that we used is based on the click through logs from a commercial search engine (google.co.in) collected over a period of two months, including totally 2,300 different queries, 2.5 million single sessions and 2.93 million clicks. On average, each query has 1,087 single sessions and 1,274 clicks. However, these queries are chosen randomly and they have totally different click numbers. Excluding those queries with less than five different clicked URLs, we still have 1,520 queries. Before using the data sets, some pre-processes are implemented to the click-through logs including enriching URLs and term processing.

When clustering feedback sessions of a query, we try five different K(1,2,…5) in K-means clustering. Then, we restructure the search results according to the inferred user search goals and evaluate the performance by CAP, respectively. At last, we select K with the highest CAP.

Before computing CAP, we need to determine $\gamma$ in (10). We select 20 queries and empirically decide the number of user search goals of these queries. Then, we cluster the feedback sessions and restructure the search results with inferred user search goals. We tune the parameter _ to make CAP the highest when K in K-means accord with what we expected for most queries. Based on the above process, the optimal $\gamma$ is from 0.6 to 0.8 for the 20 queries. The mean and the variance of the optimal $\gamma$ are 0.697 and 0.005, respectively. Thus, we set $\gamma$ to be 0.7. Moreover, we use another 20 queries to compute CAP with the optimal $\gamma$ (0.7) and the result shows that it is proper to set $\gamma$ to be 0.7. In the following, we will give the comparison between our method and the other two methods in restructuring web search results.

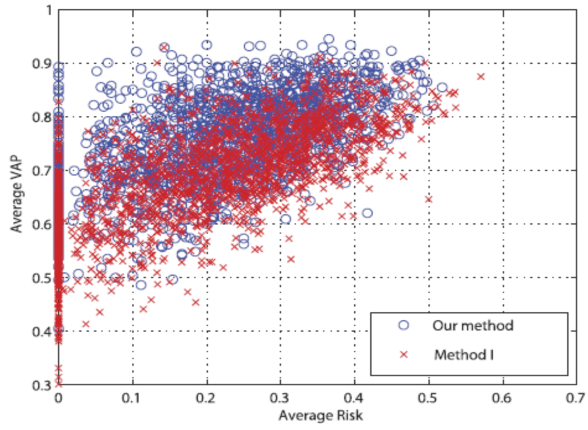
## VI. Object Evaluation and Comparison

In this section, we will give the objective evaluation of our Search goal inference method and the comparison with other two methods.

Three methods are compared. They are described as follows:

- Our proposed method clusters feedback sessions to infer user search goals.
- Method I clusters the top 100 search results to infer user search goals [6], [20]. First, we program to automatically submit the queries to the search engine again and crawl the top 100 search results including their titles and snippets for each query. Then, each search result is mapped to a feature vector according to (1) and (2). Finally, we cluster these 100 search results of a query to infer user search goals by K-means clustering and select the optimal K based on CAP criterion.
- .Method II clusters different clicked URLs directly [18]. In user click-through logs, a query has a lot of different single sessions; however, the different clicked URLs may be few. First, we select these different clicked URLs for a query from user click through logs and enrich them with these titles and snippets as we do in our method. Then, each clicked URL is mapped to a feature vector according to (1) and (2). Finally, we cluster these different clicked URLs directly to infer user search goals as we do in our method and Method I.

In order to demonstrate that when inferring user search goals, clustering our proposed feedback sessions are more efficient than clustering search results and clicked URLs directly, we use the same framework and clustering method. The only difference is that the samples these three methods cluster are different. Note that in order to make the format of the data set suitable for Method I and Method II, some data reorganization is performed to the data set. The performance evaluation and comparison are based on the restructuring web search results.



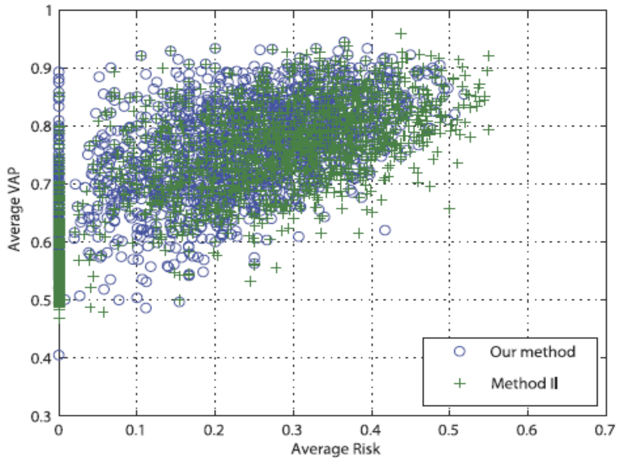




Fig. 8. Comparison of three methods for 1,520 queries. Each point represents the average Risk and VAP of a query when evaluating the performance of restructuring the search results.

As shown in Fig. 8, we compare three methods for all the 1,520 queries. Fig. 8a compares our method with Method I and Fig. 8b compares ours with Method II. Risk and VAP are used to evaluate the performance of restructuring search results together.

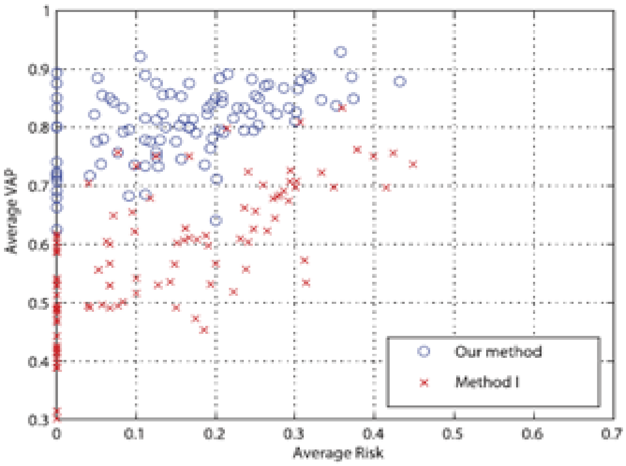
Each point in Fig. 8 represents the average Risk and VAP of a query. If the search results of a query are restructured properly, Risk should be small and VAP should be high and the point should tend to be at the top left corner. We can see that the points of our method are closer to the top left corner comparatively. We compute the mean average VAP, Risk, and CAP of all the 1,520 queries as shown in Table 2.

We can see that the mean average CAP of our method is the highest, 8.22 and 3.44 percent higher than Methods I and II respectively. The results of Method I are lower than ours due to the lack of user feedbacks. However, the results of Method II are close to ours.

TABLE 2
CAP Comparison of Three Methods for 1,520 Queries

| Method | Mean Average VAP | Mean Average Risk | Mean Average CAP |
|---|---|---|---|
| Our Method | 0.755 | 0.224 | 0.632 |
| Method I | 0.680 | 0.196 | 0.584 |
| Method II | 0.742 | 0.243 | 0.611 |

Below are the experimental results for comparison of three methods for 100 most ambiguous queries as shown in Fig (9)




(a)

Fig. 9. Comparison of three methods for 100 most ambiguous queries.

Each point represents the average Risk and VAP of a query when evaluating the performance of restructuring the search results.

TABLE 3
CAP Comparison of Three Methods for 100 Most Ambiguous Queries

| Method | Mean Average VAP | Mean Average Risk | Mean Average CAP |
|---|---|---|---|
| Our method | 0.807 | 0.159 | 0.715 |
| Method I | 0.583 | 0.138 | 0.525 |
| Method II | 0.750 | 0.231 | 0.624 |

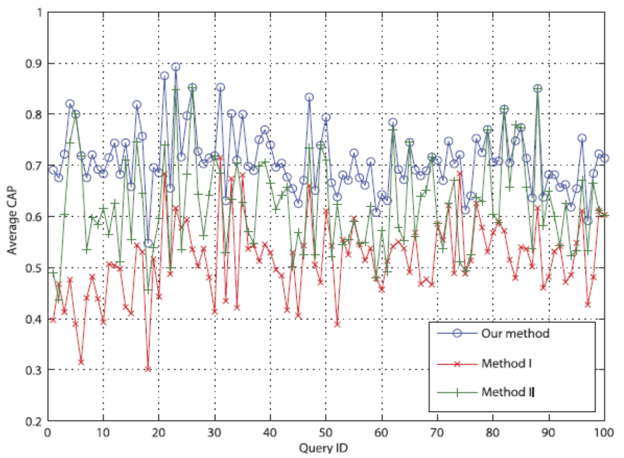



Fig 10.The Chart of CAP comparison of three methods for 100 most ambiguous queries

## VII. ADVANTAGES

- First, we can restructure web search results ac-cording to user search goals by grouping the search results with the same search goal; thus, users with different search goals can easily find what they want.
- Second, user search goals represented by some keywords can be utilized in query recommendation thus, the suggested queries can help users to form their queries more precisely.
- Third, the distributions of user search goals can also be useful in applications such as reranking web search results that contain different user search goals.

## VIII. CONCLUSION

In this paper, a new metric has been proposed to infer user search goals for a query by clustering its feedback sessions represented by pseudo-documents. First, we introduce feedback sessions to be analyzed to infer user search goals rather than search results or clicked URLs. It contains clicked URLs and the unclicked ones before the last click and corresponding page rank are considered as user implicit feedbacks and taken into ac-count to construct feedback sessions. Therefore, feedback sessions can reflect user information needs more efficiently.

Second, we map feedback sessions to pseudo documents to approximate goal texts in user minds. The pseudo-documents can enrich the URLs with additional textual contents including the titles and snip-pets. Based on these pseudo-documents, user search goals can then be discovered and depicted with some keywords. Finally, a
performance of user search goal inference. The complexity of our approach is low and our approach can be used in reality easily. For each query, the running time depends on the number of feedback sessions.

## REFERENCES


[1] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. ACM Press, 1999. 512 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 3, MARCH 2013, Clustering of the clicked URLs and feedback sessions, where points represent clicked URLs, ellipses represent feedback sessions, and lines represent class boundaries. Supposing that the clicked URLs have two classes, the points in the left figure are hard to be segmented directly, while the ellipses in the right figure are easier to be segmented.

[2] R. Baeza-Yates, C. Hurtado, and M. Mendoza, "Query Recommendation Using Query Logs in Search Engines," Proc. Int'l Conf. Current Trends in Database Technology (EDBT '04), pp. 588-596, 2004.

[3] D. Beeferman and A. Berger, "Agglomerative Clustering of a Search Engine Query Log," Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '00), pp. 407-416, 2000.

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

[5] 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 Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '08), pp. 875-883, 2008. [6] H. Chen and S. Dumais, "Bringing Order to the Web: Automatically Categorizing Search Results," Proc. SIGCHI Conf. Human Factors in Computing Systems (SIGCHI '00), pp. 145-152, 2000.

[7] C.-K Huang, L.-F Chien, and Y.-J Oyang, “Relevant Term Suggestion in Interactive Web Search Based on Contextual Information in Query Session Logs,” J. Am. Soc. for Information Science and Technology, vol. 54, no. 7, pp. 638-649, 2003.

[8] T. Joachims, “Evaluating Retrieval Performance Using Clickthrough Data,” Text Mining, J. Franke, G. Nakhaeizadeh, and I. Renz, eds., pp. 79-96, Physica/Springer Verlag, 2003.

[9] T. Joachims, “Optimizing Search Engines Using Clickthrough Data,” Proc. Eighth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’02), pp. 133-142, 2002.

[10] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay, “Accurately Interpreting Clickthrough Data as Implicit Feedback,” Proc. 28th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’05), pp. 154-161, 2005.

[11] 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.

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

[13] U. Lee, Z. Liu, and J. Cho, “Automatic Identification of User Goals in Web Search,” Proc. 14th Int’l Conf. World Wide Web (WWW ’05), pp. 391-400, 2005.

[14] X. Li, Y.-Y Wang, and A. Acero, “Learning Query Intent from Regularized Click Graphs,” Proc. 31st Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’08), pp. 339-346, 2008.

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

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

[17] D. Shen, J. Sun, Q. Yang, and Z. Chen, “Building Bridges for Web Query Classification,” Proc. 29th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’06),pp. 131-138, 2006.

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

[19] J.-R Wen, J.-Y Nie, and H.-J Zhang, “Clustering User Queries of a Search Engine,” Proc. Tenth Int’l Conf. World Wide Web (WWW ’01), pp. 162-168, 2001.

[20] H.-J Zeng, Q.-C He, Z. Chen, W.-Y Ma, and J. Ma, “Learning to Cluster Web Search Results,” Proc. 27th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’04), pp. 210-217, 2004.

**M.Monika,**M.Tech, Mallareddy engineering college for women,, Bahadurpally, Rangareddy Dist (TS), India

**N.Rajesh,** Asst.Professor, Mallareddy engineering college for women,, Bahadurpally, Rangareddy Dist (TS), India

**K.Rameshbabu,**Head of CSE depart, Mallareddy engineering college for women,, Bahadurpally, Rangareddy Dist (TS), India