Inferring User Search goals Engine Using Bisecting Algorithm

Deepali Agale,Beena Khade

Abstract— Different users may have different search goals when they submit broad-topic and ambiguous query, to a search engine. The inference and analysis of user search goals can be very useful in improving performance of search engine. To infer user search goals by analyzing search engine query logs a novel approach is proposed. First, we propose a framework to find out different user search goals for a query by clustering the proposed feedback sessions. Feedback sessions are built from user click-through data and can efficiently reflect the information needs of users. Second, then propose a novel approach to generate pseudo-documents by using feedback sessions for clustering. For clustering we use a new algorithm which is bisecting K-means algorithm. At the end, a new criterion "Classified Average Precision (CAP)" is proposed to evaluate the performance of inferring user search goals.

Index Terms—User Search Goals Feedback Sessions, Pseudo-Documents, Restructuring Search Results, Classified Average Precision

1 INTRODUCTION

Many ambiguous queries may cover a broad topic and different users may want to get information about different aspects when they submit the same query. For example, when the query "the sun" is submitted to a search engine, some users want to get information about a United Kingdom newspaper, while some others want to learn about the natural knowledge of the sun.

In this paper, we aim at searching the number of diverse user search goals for a query and depicting each goal with some keywords automatically. First we propose a new approach to infer user search goals for ambiguous query by clustering our proposed feedback sessions. The feedback session is defined as the series of clicked and unclicked URLs and ends with the last URL that was clicked in a session from user click-through logs. Then, we propose an optimization method to map feedback sessions to pseudo-documents. Then, we cluster these pseudo documents to infer user search goals and depict them with some keywords. Since, we also propose an evaluation criterion classified average precision (CAP) to evaluate the performance of the restructured web search results.

2 MOTIVATION

User may have different information needs according to his queries. This project is basically used for broad topic and ambiguous queries because ambiguous queries mean that same word has multiple results. Hence when user want to get information about ambiguous query it displays the different results. The inference and analysis of user search goals can be very useful in improving search engine relevance and user experience

3 RELATED WORK

Many works about user search goals analysis have been investigated. This can be summarized into two classes: classification of query, search result reorganization.

In this first class, in paper [1] topical classification of web queries has drawn recent interest because of the promise it offers in improving retrieval effectiveness and efficiency. However, much of this promise depends on whether classification is performed before or after the query is used to retrieve documents. In the second class, in paper [2] Effective organization of search results is critical for improving the utility of any search engine. Clustering of search results is an effective way to organize search results, that allows a user to navigate into relevant documents quickly. However, there are two deficiencies of this approach: (1) the clusters discovered do not necessarily correspond to the interesting aspects of a topic from the user's perspective; and (2) the cluster labels generated are not informative enough to allow user to identify the right cluster. In this paper, we propose to address these two deficiencies by (1) learning \interesting aspects" of a topic from Web search logs and organizing search results accordingly; and (2) generating more meaningful cluster labels using past query words entered by users. We try to evaluate our proposed method on a commercial search engine data. Compared with the traditional methods of clustering of search results, our method can obtain better result organization and more meaningful labels. People try to reorganize search results. But this involves many noisy search results that are not clicked by any users. In the third class, people aim at detecting session boundaries. However, this only identifies whether pair of queries belongs to the same goal and does not care what the goal is in detail. In paper[7] author tried to cluster pseudo document by using clustering but to improve performance of system we are trying to implement a new bisecting K-means clustering algorithm.

International Journal of Scientific & Engineering Research Volume 4, Issue 12, December-2013
ISSN 2229-5518

Table1. Previous Work

PREVIOUS RESEARCH PAPERS	RESULT/CONCLUSION
Z. Chen	Worked on Query classification Limitations- Experiment was conducted on a potentially-biased dataset
H. Chen	Organizes search results into a hierarchical category structure. Limitations- Query aspects without user feedback have limitations to improve search engine relevance
Wang , Zhai	clustered queries and learned aspects of similar queries Limitations- This method does not work if we try to discover user search goals of any one single query in the query cluster rather than a cluster of similar queries.
R. Jones and K.L. Klinkner,	Introduce search goals and missions to detect session boundary hierarchically Limitations- Their method only identifies whether a pair of queries belong to the same goal or not and does not care what the goal is in detail.

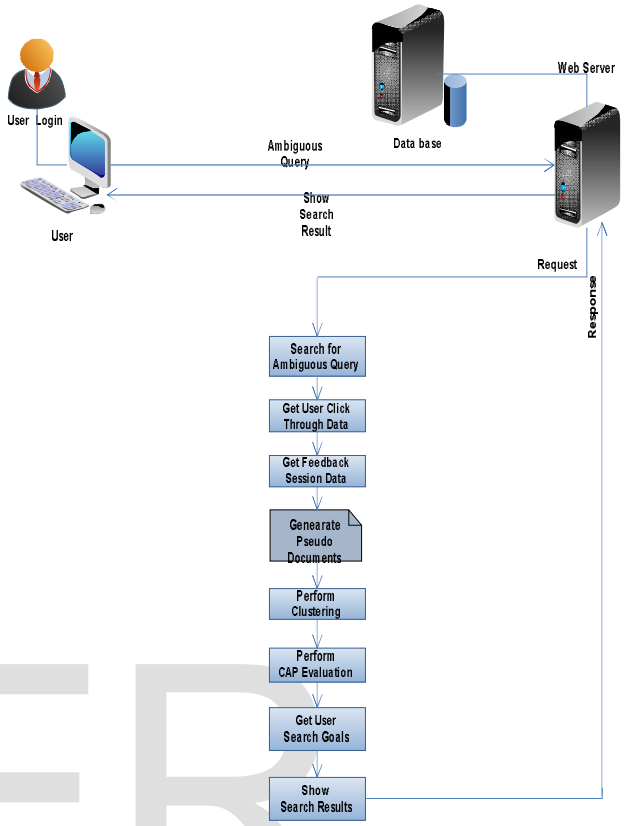

Figure.1 System Architecture

4 PROPOSED WORK

Overall system architecture is as shown in figure 1. Initially user login into system and search for ambiguous query. Then system shows many search results. From this results user clicked some desired URLs, by using this clicked data system makes a feedback session. After that feedback session is mapped to pseudo document and clustering is performed. At last performance of system is calculated by using CAP evaluation criteria.

Here, we first describe the proposed feedback sessions and then we introduce the proposed pseudo-documents to represent feedback sessions.

4.1 FEEDBACK SESSION

The proposed feedback session consists of both clicked and unclicked URLs and ends with the last URL that was clicked in a single session. It is motivated that before the last click of URL, all the URLs have been scanned and evaluated by the users. Therefore, besides the clicked URLs, the unclicked ones before the last click should be a part of the user feedbacks. Fig. 4.1 shows an example of a feedback session and a single session.

International Journal of Scientific & Engineering Research Volume 4, Issue 12, December-2013
ISSN 2229-5518

Search results	Click sequence
www.thesun.co.uk/	0
www.nineplanets.org/sol.html	1
www.solarviews.com/eng/sun.htm	2
en.wikipedia.org/wiki/Sun	0
www.thesunmagazine.org/	0
www.space.com/sun/	0
en.wikipedia.org/wiki/The_Sun_(newspaper)	3
imagine.gsfc.nasa.gov/docs/science/know_l1/sun.html	0
www.nasa.gov/worldbook/sun_worldbook.html	0
www.enchantedlearning.com/subjects/astronomy/sun/	0

Figure.2 Feedback session in single session

In Fig.1, the left part lists 10 search results of the query "the sun" and the right part is a user's click sequence where "0" means "unclicked." The single session includes all the 10 URLs in Figure.2, while the feedback session only includes the seven URLs in the upper rectangular box. Out of seven URLs three are clicked URLs and four are unclicked URLs in this example.Since users will scan the URLs one by one from top to bottom, we can assume that besides the three clicked URLs, the four unclicked URLs in the rectangular box have also been browsed and evaluated by the user and they should be a part of the user feedback. In the part of feedback session, the clicked URLs tell what users require and the unclicked URLs reflect what users do not care about. It should be noted that the unclicked URLs after the last clicked URL should not be included into the feedback sessions since it is not certain whether they were scanned or not. Each feedback session can tell what a user requires and what he/she does not care about. Moreover, there are lots of diverse feedback sessions in user click-through searched results and clicked URLs logs. Therefore, for inferring user search goals, it is more efficient to analyze the feedback sessions than to analyze the search results and clicked URLs.

4.2 CONVERSION OF FEEDBACK SESSIONS TO PSEUDO-DOCUMENTS

Building of pseudo-document has two steps.

4.2.1 Representing the URLs In the Feedback Session.

In the first step, we first enrich the URLs with additional textual contents by extracting the titles and snippets of the returned URLs appearing in the feedback session. In such way, each URL in the feedback session is represented by a small text paragraph that consists of that URLs title and snippet. After that some textual processes are implemented to those text paragraphs, such as transforming all of the letters to lowercases, stemming and removing stop words. Finally, each URL's title and snippet are represented by a Term Frequency-Inverse Document Frequency (TF-IDF) vector, respectively, as in

$$Ti = [tw_1, tw_2, tw_3 \ldots\ldots twn]^T$$
$$Si = [Sw_1, Sw_2, Sw_3 \ldots\ldots Swn]^T \quad (1)$$

Where Ti and Sj are the TF-IDF vectors of the URL's title and snippet, respectively. wj= (1,2,3,……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. twj and swj represent the TF-IDF value of the jth term in the URL's title and snippet, respectively. Considering that each URLs' titles and snippets have different significances, we represent the each enriched URL by the weighted sum of Tui and Sui, namely

$$Ri = wt \cdot \mathbf{T}i + st \cdot \mathbf{S}i = [f\,w_1, f\,w_2, \ldots\ldots\ldots\ldots, fwn]^T \quad (2)$$

Where Ri means the feature representation of the i^{th} URL in the feedback session, and *w*t and *s*t are the weights of the titles and the snippets, respectively.

4.2.2 Formation of Pseudo-Document

We propose an optimization method to combine clicked and unclicked URLs in the feedback session to obtain a feature representation.

Let R be the feature representation of a feedback session, and (*w*) be the value for the term w.

Let

C = (m=1, 2, 3… M), and

UC = (l=1, 2, 3… L);

Let R be the feature representations of the clicked and unclicked URLs in this feedback session, respectively.

Let C and UC be the values for the term w in the vectors. We want to obtain such a S that the sum of the distances between S and each C is minimized and the sum of the distances between S and each UC 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 below equation.

$$S = [ff(w\square), ff(w\square), ff(w\square), \ldots\ldots\ldots, ff(wn)]\square \quad (3)$$

International Journal of Scientific & Engineering Research Volume 4, Issue 12, December-2013
ISSN 2229-5518

$$Rs(w) = \arg\min \sum_{M} \left(S(w) - C(w)\right)^2 - \lambda \sum_{L} \left((S(w) - UC(w)\right)^2 \quad (4)$$

λ is a parameter balancing the importance of clicked and unclicked URLs. When λ in (4) is 0, unclicked URLs are not taken into account. On the other hand, if $\lambda$is too big, unclicked URLs will dominate the value of Uc. In this project, we set λ to be 0.5.

4.3 CLUSTERING OF PSEUDO DOCUMENT

As in equation (3) and (4), each feedback session is represented by a pseudo-document and the feature representation of the pseudo-document is Rs. The similarity between two pseudo-documents is computed as the cosine score of Rs_i and Rsj, as follows:

$$Simij = \cos(Rs_i, Rsj) = \frac{Rs_i . Rsj}{|Rs_i| . |Rsj|} \quad (5)$$

And distance between two feedback sessions is calculated by using formula

$$Disij = 1 - Simij$$

To cluster pseudo documents K-means clustering is used which is very simple and effective. To check the optimal values of clustering we have a evaluation criterion.

4.4 BISECTING ALGORITHM

For Bisecting algorithm you must cluster documents using k-means algorithm and then on the result of k-means algorithm you can apply bisecting algorithm.

Read following bisecting steps.
The idea is iteratively splitting your cloud of points in 2 parts. In other words, you build a random binary tree where each splitting (a node with two children) corresponds to splitting the points of your cloud in
You begin with a cloud of points.

- Compute its centroid (barycenter) w
- Select randomly a point cL among the points of the cloud
- Construct point cR as the symmetric point of cL when compared to w (the segment cL->w is the same as w->cR)
- Separate the points of your cloud in two, the ones closest to cR belong to the subcloud R, and the ones closest to cL belongs to the subcloud L
- Reiterate for the subclouds R and L

Notes :
You can discard the random points once you've used them already. However, keep the centroids of all the subcoulds.
Stop at point when your subclouds contain exactly one point.

5. EVALUATION CRITERION

5.1 AVERAGE PRECISION

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

$$AP = \frac{1}{N^+} \sum_{r=1}^{N} rel(r) \frac{R_r}{r} \quad (6)$$

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(r) 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.

5.2 VOTED AP (VAP)

It is calculated for purpose of restructuring of search results classes i.e. different clustered results classes. It is same as AP and calculated for class which having more clicks.

5.3 RISK

It is the AP of the class including more clicks? There should be a risk to avoid classifying search results into too many classes by error. So we propose the Risk.

$$Risk = \frac{\sum_{i,i=1(i<j)}^{m} dij}{C^2 m}$$

5.4 CLASSIFIED AP (CAP)

VAP is extended to CAP by introducing combination of VAP and Risk. Classified AP can be calculated by using the formula, as follows:

$$CAP = VAP \times (1 - Risk)$$

6. CONCLUSION

The proposed system can be used to improve discovery of user search goals for a similar query by using bisecting algorithm for clustering user feedback sessions represented by pseudo-documents. By using proposed system, the inferred user search goals can be used to restructure web search re-

International Journal of Scientific & Engineering Research Volume 4, Issue 12, December-2013
ISSN 2229-5518

sults. So, users can find exact information quickly and very efficiently. The discovered clusters of query can also be used to assist users in web search.

ACKNOWLEDGMENT

Ms. Deepali Agale is thankful to Prof. B.S.Khade, Asst. Professor, Information Technology Department, Bhivarabai Sawant Institute of Technology and Research, Wagholi, Pune, for her constant support and helping out with the preparation of this paper. Also thankful to the Principal, Mr.D.M.Yadav Bhivarabai Sawant Institute Of Technology and Research ,Wagholi,Pune, and Prof. G.M.Bhandari, HOD, Computer Engineering Department Bhivarabai Sawant Institute Of Technology and Research ,Wagholi,Pune, for being a constant source of inspiration.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.