A Survey on: A New Algorithm for Inferring User Search Goals with Feedback Sessions

Swati S. Chiliveri, Pratiksha C.Dhande

Abstract— **For a broad-topic and ambiguous query, different users may have different search goals when they submit it to a search engine. The inference and analysis of user search goals can be very useful in improving search engine relevance and user experience. This paper provides an overview of the system architecture of proposed feedback session framework with their advantages. Also we have briefly discussed the literature survey on Automatic Identification of User Goals in Web Search with their pros and cons. We have presented working of various classification approaches such as SVM, Exact matching and Bridges etc. under the web query classification framework. Lastly, we have described the existing web clustering engines such as MetaCrawler, Carrot 2, iBoogie,Vivisimo etc.**

Index Terms— **Classified Average Precision(CAP), Feedback Sessions, Open Directory Project (ODP) , Pseudo-documents, Restructuring search results, User search goals s.**

I. INTRODUCTION

In web search applications, user submits the queries to search engines to represent the information needs. However, sometimes queries that are submitted to search engine may not exactly represent users' specific information needs .Since many ambiguous queries may cover a broad topic and different users may want to get the different information on different aspects when they submit the same query. Now 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 difficult 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 "the sun" is submitted to a search engine, some people want to locate the homepage of a United Kingdom newspaper, while some people want to learn the natural knowledge of the sun. Therefore, it is necessary and potential to capture different user search goals, 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 users need. 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, we have listed Some advantages.
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 reranking web search results that contain different user search goals.

Manuscript received Nov, 2014.

Swati S. Chiliveri, *ME(Computer Engg),Savitribai Phule University Pune , Maharashtra(INDIA), Phone/ Mobile No:09021856445*
Pratiksha C.Dhande, *M Tech(Computer Science),Savitribai Phule University Pune , Maharashtra(INDIA), Phone/ Mobile No:08600993439.*

ISSN: 2278 – 1323

A. System Architecture

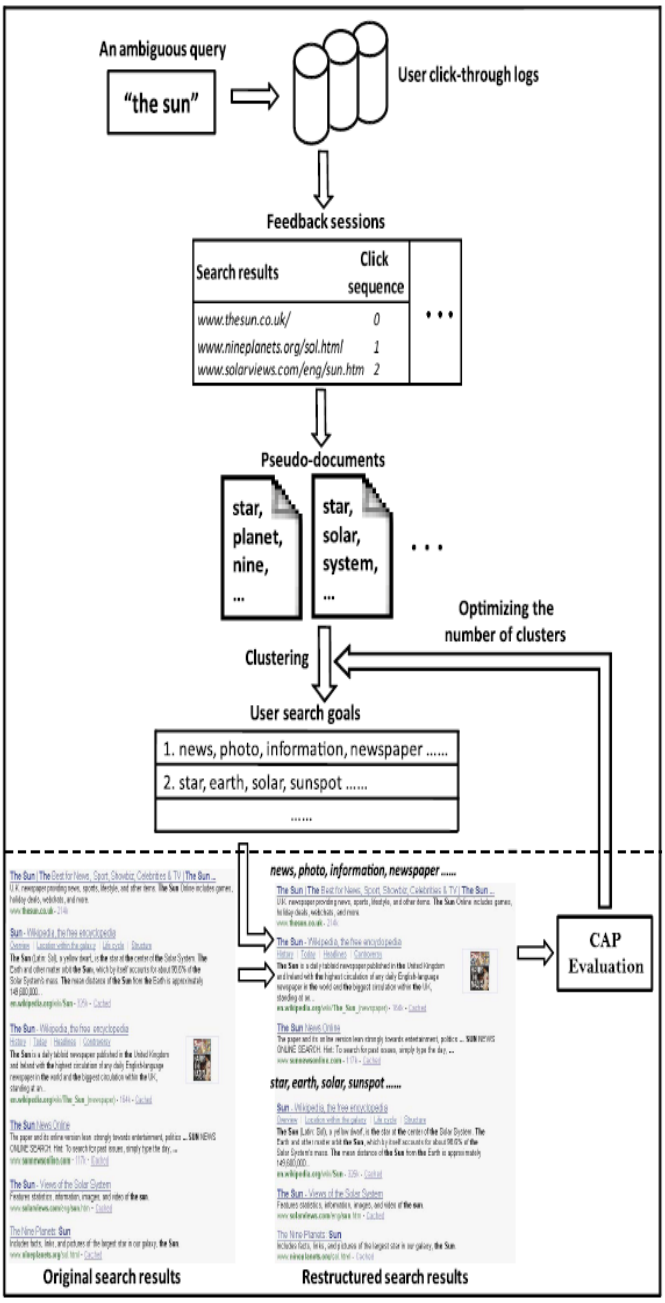

Figure 2.1 Inferring User Search Goals with Feedback Sessions

Fig. 2.1 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 evaluate 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..

II.LITERATURE SURVEY:

A. Automatic Identification of User Goals in Web Search

Here, Author propose two types of features for the goal-identification task:
1. User-Click Behavior
2.Anchor-Link Distribution.

1.Past User Click Behaviour:

a. Click Distribution.

In this work [2], Author suggested that the user's goal for a given query can be learned from how users in the past have interacted 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.

On the other hand, if the goal is informational, in the past users should have clicked on many results related to that query. In this way just by observing how the results for a particular query have been clicked so far, we can come to know that whether the current user who issues that query has a navigational or an informational goal.

b. Average Number Of Clicks Per Query.

Besides click distribution, We need to focus on another feature embedded in the user-click behavior is how many results a user clicks on after 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. On the other hand, for an informational query, the user may click on several results. Therefore, Author use the number of clicks per query as another potential feature based on user-click behavior.

2.Anchor Link Distribution

For a given a query, its anchor-link distribution is computed as follows:

Firstly, locate all the anchors appearing on the Web that have the same text as the query, and extract their destination URL's. Then, count how many times each destination URL appears in this list .After getting the count of destination URL , sort the destinations in the descending order of their appearance. Then author created a histogram where the frequency count in the I^{th} bin is the number of times that the I^{th} destination appears. Finally, normalize the frequency in each bin so that all frequency values add up to 1.

For a navigational query, because of the existence of an authoritative answer, author expect the anchor-link distribution to be highly skewed toward rank one . On the other hand, the anchor-link distribution for an informational query should be more flat because of the lack of consensus regarding which Website provides the most authoritative answer.

3.Advantages And Limitation:

This proposed features are much more effective than the term-occurrence pattern-based features.

It is possible to get 90% accurate results in goal identification

One limitation of this approach is that, experiment was conducted on a potentially-biased dataset: queries from the CS department may show a technical bias and are likely to be well crafted and potentially work related.

B. Building Bridges for Web Query Classification

In this paper [3], author present a novel approach for query classification that outperforms the winning solution of the ACM KDDCUP 2005 competition, whose objective is to classify 800,000 real user queries. Author, 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 categories via the above intermediate taxonomy.

One major innovation is that author do not need to retrain a new classifier for each new set of target categories, and therefore the bridging classifier needs to be trained only once. In this work author used category selection as a new method for narrowing down the scope of the intermediate taxonomy based on which we classify the queries. Category selection can improve both efficiency and effectiveness of the online classification.

1. Classification Approaches:

a. Classification By Exact Matching

Author describe two taxonomy in this work intermediate taxonomy and the target taxonomy as

C^I and C^T respectively. For each category in C^I, detect whether it is mapped to any category in C^T. After that, the most frequent target categories to which the returned intermediate categories have been successfully mapped are regarded as the classification result.

Here, exact matching approach produce classification results with high precision but low recall. Exact matching approach produces high precision because this approach relies on the Web pages which are associated with the manually annotated category information.

It produces low recall because many search result pages have no intermediate categories.

The exact matching approach cannot find all the mappings from the existing intermediate taxonomy to the target taxonomy which also results in low recall.

b. Classification by SVM

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

Query classification with SVM consists of the following steps:

1) construct the training data for the target categories based on mapping functions between categories. If an intermediate category C^I is mapped to a target category C^T, then the Web pages in C^I are mapped into C^T;

2) Train SVM classifiers for the target categories.

3) For each Web query to be classified, use search engines to get its enriched features and classify the query using the SVM classifiers.

c. Classifiers By Bridges

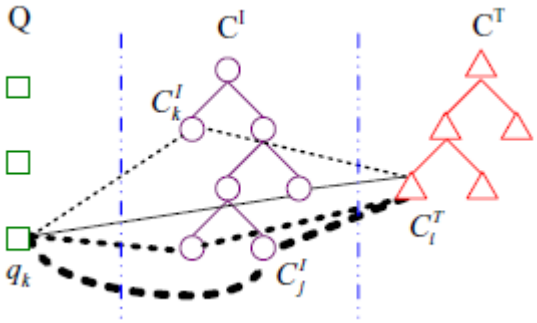

Figure3.1 Taxonomy Bridging Classifier

Here, author describe new query classification approach called taxonomy bridging classifier, or bridging classifier . It provide the connection between the target taxonomy and queries by taking an intermediate taxonomy as a bridge. The above figure shows the working of taxonomy bridging classifier, where two vertical lines separate the space into three parts. The square in the left part denotes the queries to be classified; the tree in the right part represents the target taxonomy; the tree in the middle part is an existing intermediate taxonomy. The thickness of the dotted lines reflects the similarly relationship between two nodes.

In above figure, we can easily identify that the relationship between C_i^T and C_j^I is much stronger than that between C_i^T and C_k^I. Given a category C_i^T in the target taxonomy and a query to be classified q_k, we can judge the similarity between them by the distributions of their relationship to the categories in the intermediate taxonomy.

2. Advantages And Limitation

In this work ,an intermediate taxonomy is used to train classifiers bridging the queries and target categories so that there is no need to collect the training data.

This approach suffers from two potential problems.

1.The classifier for the second mapping function needs to be trained whenever the target category structure changes. Since in real applications, the target categories can change depending on the needs of the service providers, as well as the distribution of the Web contents, this solution is not flexible enough. It is better to train the classifiers only once and then use this in future for the query classification tasks, even when the target categories are different.

2. Author used the Open Directory Project (ODP) taxonomy as the intermediate taxonomy. Since the ODP contains more than 590,000 different categories, it is very costly to handle all mapping functions. It is better to select a portion of the most relevant parts of the intermediate categories.

ISSN: 2278 – 1323

C. Learn from Web Search Logs to Organize Search Results

For a given input query to the search engine , the general procedure of this proposed approach is described as follows[4]:
1. Get its related information from search engine logs. All the information forms a working set.
2. Learn aspects from the information in the working set. These aspects correspond to users' interests given the input query. Each aspect is labeled with a representative query.
3. Categorize and organize the search results of the input query according to the aspects learned above.

1. Finding Related Past Queries

Given a query q, 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 retrieve its past similar queries from preprocessed history data collection.
Assume that there are N pseudo-documents in history data set: $H = \{Q_1 , Q_2 , ..., Q_N \}$. Each Q_i corresponds to a unique query and is enriched with clickthrough information .
To find q's related queries in H, a natural way is to use a text retrieval algorithm.
Here author have used the OKAPI method ,it is one of the state-of-the art retrieval method.
After calculating the similarity between query q and pseudo-document Qi , 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 corresponds to a past query, and thus the top ranked documents correspond to q's related past queries.

2. Learning Aspects by Clustering

Given a query q, $Hq = \{d_1 , ..., d_n \}$ represent the set of top ranked pseudo-documents from the history 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. In this work , author have 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.
In the past query collection Hq , each document corresponds to a query. This center query can be regarded as the most representative one for the whole cluster, and thus provides a label for the cluster naturally. All the clusters obtained are related to the input query q from a different perspective, and they represent the possible aspects of interests about query q of users.

3. Categorizing 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 q: $\{s_1 , ..., s_m \}$, group them into different aspects using a categorization algorithm.
Here, author have used a simple centroid based method for categorization.

4. Advantages And Limitation:

1. Comparison of this proposed log-based method with the traditional cluster-based method and the baseline of search engine ranking, the experimental results clearly show that log-based
method can consistently outperform cluster-based method and improve over the ranking baseline, especially when the queries are difficult or the search results are diverse.
2. Log-based method can generate more meaningful aspect labels than the cluster labels generated based on search results when we cluster search results.

D. Grouper: A Dynamic Clustering Interface To Web Search Results.

In this work [5], author used Suffix Tree Clustering (STC) to identify set of documents having common phrases and then create cluster based on these phrases or contents. Here, author used documents snippets instead whole document for clustering web documents. However, generating meaningful labels for clusters is one of the most challenging task in document clustering. So, to overcome this difficulty, author used a supervised learning method to extract possible phrases from search result snippets or contents and these phrases are then used to cluster web search results.

1. 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 satisfy the stringent requirements of the Web domain. It is shown that STC is faster than standard 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 proximity information between words. Suffix Tree Clustering mainly relies on a suffix tree to identify sets of documents that share common phrases and uses this information to create clusters and summarize their contents for users successfully.

2. Advantages And Limitation of STC

1.Suffix Tree Clustering uses phrases to provide concise and meaningful descriptions of groups. 2.STC's thresholds play a significant role in the process of cluster formation, and they turn out particularly difficult to tune.

3.STC's phrase pruning heuristic tends to remove longer high quality phrases, leaving only the shorter and less informative ones.
4.If a document does not include any of the extracted phrases or just some parts of them, it will not be included in the results although it may still be relevant.

III.EXISTING WEB CLUSTERING ENGINES

A. MetaCrawler: A Meta Search Engine

MetaCrawler is a metasearch engine that blends the top web search results from Google, Yahoo!, Bing (formerly Live Search), Ask.com, About.com, MIVA, LookSmart and other popular search engines. MetaCrawler also provides users the option to search for images, video, news, yellow pages and white pages. MetaCrawler also provide the option to search for audio.

MetaCrawler was originally developed in 1994 at the University of Washington Erik Selberg and Professor Oren Etzioni as Selberg's Ph.D project. Originally, it was created in order to provide a reliable abstraction layer to early Web search engines such as WebCrawlers,Lycos and InfoSeek in order to study semantic structure on the Web.

B. Carrot2: A Web Clustering Engine

Carrot2 combines several search results clustering algorithms: STC, Lingo, TRSC, clustering based on swarm intelligence (ant-colonies), and simple agglomerative Techniques. Lingo uses SVD as the primary mechanism for cluster label induction.
The system architecture of Carrot 2 is based on processing components arranged into pipelines. There are two major groups or processing components in Carrot 2 are:

1.Document Source

2.Clustering Algorithm

Document sources

Document sources generally provide data for further processing. For example, fetch the search results from an external search engine, or load text files from a local disk.

Currently, Carrot 2 has built-in support for the following document sources:

- Bing Search API
- Lucene index
- OpenSearch
- PubMed
- Solr server
- eTools metasearch engine

Clustering Algorithms

Carrot 2 offers two specialized document clustering algorithms that play important role to provide the quality of cluster labels:

- Lingo- a clustering algorithm based on the Singular value decomposition
- STC:Suffix Tree Clustering

Other algorithms can be easily added to Carrot 2.

C. Vivisimo

Vivisimo was founded by research computer scientists at the Computer Science Department at Carnegie Mellon University, where research was originally done under grants from the National Science Foundation. The company was founded in June 2000. The document clustering algorithm and meta- search software from Vivisimo automatically categorizes search results on-the-fly into hierarchical clusters. Vivisimo Velocity is built on a modern architecture and takes advantage of XML and XSL standards.

D.iBoogie: A Web Document Clustering Engine

iBoogie is a free search site developed and owned by CyberTavern. IBoogie combines metasearch and clustering algorithm to deliver and organize search results from multiple sources into structured content. This is done dynamically in real time and presented to a user in a hierarchy of topics for browsing and exploring.
The main page of iBoogie has a single search form, and users can easily select different search tabs in order to search different document types or topics (Web, Directory, Images, News, Patents, Gov, Jobs, Medical, Cars-Autos, Cooking-Recipes, Immigration). Users can also create "Custom Tab" by choosing which search engines or websites to search through simultaneously.

IV. ADVANTAGES OF CLUSTERING FEEDBACK SESSIONS

1) Feedback sessions can be considered as a process of resampling.
Generally, if we go through the original URLs in the search results as original samples, then feedback sessions can be viewed as the "processed" or "resampled" samples. These resampled samples are different from the original samples and that reflects the user information needs. Without resampling, there could be many noisy URLs in the search results, which are clicked by users many time . If we cluster the search results with these noisy URLs, the performance of clustering will degrade greatly.

Here, feedback sessions actually "resample" the URLs and exclude those noisy ones. Therefore, Feedback session approach is much better than baseline method. Furthermore, the resampling by feedback sessions brings the information of user goal distribution to the new samples. For instance, most URLs in the search results of the query "the sun" are about the sun in nature while most feedback sessions are about the newspaper. Therefore, the introduction of feedback sessions provides a more reasonable way for clustering.

2) Feedback session is also a meaningful combination of several URLs. Therefore, it can reflect user information need more precisely and there are plenty of feedback sessions to be analyzed.

ISSN: 2278 – 1323

Feedback sessions can also be viewed as a preclustering of the clicked URLs for a more efficient clustering. Moreover, the number of the combinations of the clicked URLs can be much larger
than the one of the clicked URLs themselves. Therefore, this method is better than cluster based method.

V. CONCLUSION

In this paper, a novel approach has been proposed to infer user search goals for a query by clustering its feedback sessions represented by pseudo-documents. In reality, the Feedback session can discover user search goals for some popular queries offline at first. Then, when users submit one of the queries, the search engine can return the results that are categorized into different groups according to user search goals online. Thus, users can find what they want conveniently and also satisfy the user goals. 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] Zheng Lu, Hongyuan Zha, Xiaokang Yang," A New Algorithm for Inferring User Search
Goals with Feedback Sessions", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 3, March 2013

[2] Lee. U, 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.

[3] Shen. D, 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] Wang. X 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.

[5] Zamir, O. And Etzioni, O. 1999. Grouper: A dynamic clustering interface to Web search results. Comput. Netw.31, 11–16, 1361–1374.

[6] Li. X, 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.

[7] Poblete. B 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.

[8] Wen J-R, J.-Y Nie, and H.-J Zhang, "Clustering User Queries of aSearch Engine," Proc. Tenth Int'l Conf. World Wide Web (WWW '01),pp. 162-168, 2001.

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

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

