iJRASET
International Journal For Research in
Applied Science and Engineering Technology

Volume: 3 Issue: IV Month of publication: April 2015

DOI:

www.ijraset.com *Volume 3 Issue IV, April 2015*
IC Value: 13.98 *ISSN: 2321-9653*

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Enhance web search results using user feedback sessions

S.R.Sri Abirami[#1,] Dr.C.Nalini[#2], A.P.Ponselvakumar[#3]

[#]Department of Information Technology, Kongu Engineering College, Perundurai, India

Abstract – Most search engines in use today present the user a single ordered list of documents matching the search query. Since most users do not browse past the first page of search results, this method of displaying results often limits the effectiveness of the search to the relevance of the first several documents. An alternative to a single ordered list is to cluster the search results in that cluster are then displayed in a list. Under the assumption that documents which are similar to each other are likely to be relevant to the same query, the clustering of search results are easier to browse than a single ordered list. The inference and analysis of user search goals for a query can be very useful in improving search engine relevance and user experience. A novel approach is used to infer the user search goals by analyzing the user click through logs. So we first propose a framework to find out different user search goals for a query by clustering the proposed feedback sessions. The Feedback sessions are constructed from user click-through data which can efficiently reflect the information needs of users. Then a novel approach is used to generate the pseudo documents by considering the feedback sessions. The pseudo-documents are clustered using the suffix tree clustering algorithm to restructure the search results. In order to evaluate the restructured results, we use a method called "Classified Average Precision (CAP)" to measure the performance of inferring user search goals.

Keywords-Clustering, Feedback Sessions, Pseudo-Documents

I. INTRODUCTION

Users of Web search engines are often forced to sift through the long ordered list of document "snippets" returned by the engines. The IR community has explored document clustering as an alternative method of organizing retrieval results, but clustering has yet to be deployed on most major search engines. Document clustering has long been investigated as a post-retrieval document visualization technique. Document clustering algorithms attempt to group documents together based on their similarities; thus documents relating to a certain topic will hopefully be placed in a single cluster. This can help users both in locating interesting documents more easily and in getting an overview of the retrieved document set. Numerous document clustering algorithms are available to cluster the documents and classified as hierarchical and partitional. Agglomerative Hierarchical Clustering algorithms are mostly used which successively merge the most similar objects based on the pair wise distance between objects until a termination condition hold. These algorithms are quadratic in the number of documents and are therefore too slow for our online requirements. Linear time clustering algorithms are the best candidates to comply with the speed requirement of on-line clustering. These include K-Means, Single-Pass, Buckshot and Fractionation. In this paper we have introduced another linear time algorithm STC to cluster the pseudo documents which will be briefly described in the next section. One way to cluster Web search results is to do so on the search engine.The search engine plays a major role for crawling web content on different node and organizing them into result pages, so the user can easily select the needed information by navigating through the result pages link. However, sometimes queries may not exactly represent what they want since the keywords may be polysemous or cover a broad topic and users tend to formulate short queries rather than to take the trouble of constructing long queries. For example, when the query "the jaguar" is submitted to a search engine, some users want to find the animal jaguar, while some others want to know jaguar car. Therefore, it is necessary and potential to capture different user search goals in information retrieval. The inference and analysis of user search goals can have a lot of advantages in improving search engine relevance and user experience. It is used to restructure web search results according to user search goals by grouping the search results with the same search goal. Then the user search goals represented by some keywords can be used in query recommendation; thus, the suggested queries can help users to form their queries more precisely. Finally, the distribution of user search goals can also be useful in applications such as reranking Web search results that contain different user search goals.

II. RELATED WORK

A. Previous Work on Document Clustering

In this section we review previous work on document clustering algorithms and discuss how these algorithms measure up to the requirements of the Web domain. Document clustering has been traditionally investigated mainly as a means of improving the performance of search engines by pre-clustering the entire corpus. However, clustering has also been investigated as a post-retrieval document browsing technique. Our work follows this alternative paradigm. Agglomerative Hierarchical Clustering (AHC) algorithms are probably the most commonly used. These algorithms are typically slow when applied to large document collections. Single-link and group-average methods typically take O(n2) time, while complete-link methods typically take O(n3) time. As our experiments demonstrate, these algorithms are too slow to meet the speed requirement for one thousand documents. Several halting criteria for AHC algorithms have been suggested, but they are typically based on predetermined constants. These algorithms are very sensitive to the halting criterion - when the algorithm mistakenly merges multiple "good" clusters, the resulting cluster could be meaningless to the user. In the Web domain, where the results of queries could be extremely varied (in the number, length, type and relevance of the documents), this sensitivity to the halting criterion often causes poor results. Another characteristic of the Web domain is that we often receive many outliers. This sort of "noise" reduces even further the effectiveness of commonly used halting criteria. Linear time clustering algorithms are the best candidates to comply with the speed requirement of on-line clustering. These include the K-Means algorithm - *O(nkT)* time complexity where *k* is the number of desired clusters and *T* is the number of iterations, and the Single- Pass method - *O(nK)* were *K* is the number of clusters created. One advantage of the K-Means algorithm is that, unlike AHC algorithms, it can produce overlapping clusters. Its chief disadvantage is that it is known to be most effective when the desired clusters are approximately spherical with respect to the similarity measure used. There is no reason to believe that documents (under the standard representation as weighted word vectors and some form of normalized dot-product similarity measure) should fall into approximately spherical clusters. The Single-Pass method is the most popular incremental clustering algorithm but suffers from this disadvantage, as well as from being order dependant and from having a tendency to produce large clusters. Buckshot and Fractionation are fast, linear time clustering algorithms. Fractionation is an approximation to AHC, where the search for the two closest clusters is not performed globally, but in rather locally and in a bound region. This algorithm will obviously suffer from the same disadvantages of AHC - namely the arbitrary halting criteria and the poor performance in domains with many outliers. Buckshot is a K-Means algorithm where the initial cluster centroids are created by applying AHC clustering to a sample of the documents of the collection. This sampling is risky when one is possibly interested in small clusters, as they may not be represented in the sample. Finally, we note that neither of these algorithms is incremental. In contrast to STC, all the mentioned algorithms treat a document as a set of words and not as an ordered sequence of words, thus losing valuable information. Phrases have long been used to supplement word-based indexing in IR systems. The use of lexical atoms and of syntactic phrases has been shown to improve precision without hurting recall.

B. Previous Work on feedback sessions

U. Lee, Z. Liu and J. Cho [1] proposed a method for automatic identification of user goals in web search. By analyzing the user's query, the quality of search results can be improved. In existing system, manual query log investigation was used to identify the goals. In proposed system, they used automatic goal identification process. It can use two tasks like as past user click behaviour and anchor link distribution for goal identification combining these two tasks can identify 90% goal accurately. The past user click behaviour information may be small which cannot produce efficient search results.

R.. Jones, B. Rey, O. Madani and W. Greiner [2] proposed a method of generating query substitution of new query to replace the user's original query. This technique uses modification based on query substitution. The new queries and the terms are closely related to the original queries and the terms. Query substitution is contrast with query expansion and query relaxation. The query expansion is through pseudo-relevance feedback but it is cost and lead to aimless process. The query relaxation is through Boolean or TF-IDF (Term Frequency-Inverse Document Frequency) retrieval and this reduces the specificity. The technique increases the coverage and effectiveness in the setting up of query-candidate pair but lacks in the machine translation technique.

D. Shen, J. Sun, Q. Yang, and Z. Chen [3] proposed a method of Web Query Classification. Web query classification aims to classify Web user's queries which are often short and ambiguous into a set of target categories. In this approach, the author first builds 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 by means of intermediate taxonomy. A major innovation is that by leveraging the similarity distribution over the intermediate taxonomy, the user 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.

X. Wang and C.X Zhai [4] proposed a method to organize search results from web search logs. Clustering the search results is the best way to organize the search results. By clustering the search results users finds the document quickly. There are two faults for these approaches are: 1. The clusters do not depend on the interesting aspects of users. 2. The cluster labels are not informative, so identify of right clusters is difficult. The reasons are 1. Labels are not meaningful and 2. Labels are not informative. The solution of the faults in the proposed are: 1. Learning “interesting aspects” from web search logs and organizing search results. 2. Informative cluster labels are generated using query words used by the users. Evaluation of the method is based on commercial search engine log data. Compared with traditional method to this method produce the better organization results and meaningful labels.

III. RESEARCH METHODOLOGY

In order to overcome the disadvantages that are survived under the related work, a novel approach has been introduced which is described and implemented as follows

A. *Suffix Tree Clustering*

STC is a linear time clustering algorithm (linear in the size of the document set) that is based on identifying phrases that are common to groups of documents. A phrase in our context is an ordered sequence of one or more words. We define a base cluster to be the set of documents that share a common phrase.

STC has three logical steps: (1) document “cleaning”, (2) identifying base clusters using a suffix tree, and (3) combining these base clusters into clusters which are described in the following sessions.

In the document "cleaning" step, the string of text representing each document is transformed using a light stemming algorithm. Sentence boundaries are marked and non-word tokens (such as numbers, HTML tags, and most punctuation) are stripped.

The second step – the identification of base clusters – can be viewed as the creation of an inverted index of phrases for our document set. This is done efficiently using a data structure called a suffix tree. This data-structure can be constructed in time linear with the size of the document set, and can be constructed incrementally as the documents are being read. Each base cluster is assigned a score that is a function of the number of documents it contains, and the number of words that make up its phrase.

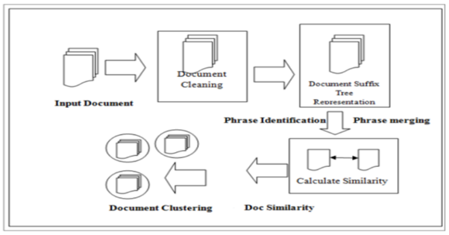


Fig. 1 Suffix tree document clustering system design

The final step of the STC algorithm merges base clusters with a high degree of overlap in their document sets. This creates clusters that are more coherent semantically by allowing documents to be in the same cluster even if they do not share a common phrase but rather share phrases with other documents of the cluster.

B. *Implementation Of Search Engine*

The Search engine environment can be created using by the integration of Apache Nutch and Apache Solr.

1) *Apache Nutch:* Apache Nutch is an open source Web crawler written in Java. It is used to find Web page hyperlinks in an automated manner, reduce lots of maintenance work, for example checking broken links, and create a copy of all the visited pages for searching over. The crawler system is driven by the Nutch crawl tool to build and maintain several types of data structures, including the web database, a set of segments, and the index.
2) *Apache Solr:* Apache Solr is a fast open-source Java search server. Solr enables you to easily create search engines which searches websites, databases and files. Its major features include full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling. It runs as a standalone full-text search server within a servlet container such as Apache Tomcat or Jetty.

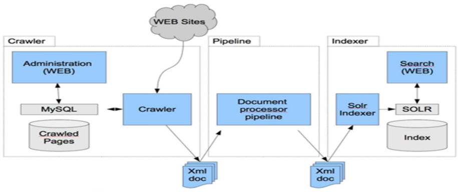


Fig. 2 Architecture of Apache Nutch and Apache Solr

3) *Integration of Apache Nutch and Apache Solr*: The contents of the index are crawl to Solr, by execute the command in the Nutch installed package runtime \ local path indexing: bin / nutch solrindex http://localhost:8983/solr/ crawl/crawldb -linkdb crawl/linkdb crawl/segments/*

C. Representation Of Feedback Sessions

The 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, all the URLs have been scanned and evaluated by users. Therefore besides the clicked URLs, the unclicked ones before the last click should be a part of the user feedbacks. The clicked URLs tell about the users need and the unclicked URLs reflect the 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 are scanned or not.

D. Mapping Feedback Session To Pseudo-Documents

Each URL in a feedback session is represented by a small text paragraph that consists of its title and snippet. Each URL's title and snippet are represented by a Term Frequency-Inverse Document Frequency (TF-IDF) vector respectively, as in Equation 1

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

where T_{ui} and S_{ui} are the TF-IDF vectors of the URL's title and snippet respectively. u_i means that ith URL in the feedback session. And w_j (j=1, 2,...n) is the jth term appearing in the enriched URLs. t_{wj} and s_{wj} represent the TF-IDF value of the jth term in the URL's title and snippet respectively. Considering that URLs titles and snippets have different significances, represent the enriched URL by the weighted sum of and T_{ui} and S_{ui} , namely

$$F_{ui} = {}_{wt} T_{ui} + {}_{ws} S_{ui} \qquad (2)$$
$$= [\, f_{w1}, f_{w2}, \ldots\ldots f_{wn}]^T$$

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 snippets respectively. It is worth noting that although T_{ui} and S_{ui} are TF-IDF features, Fui is not a TF-IDF feature. This is because the normalized TF feature is relative to the documents and therefore it cannot be aggregated across documents.

TABLE I
REPRESENTATION OF FEEDBACK SESSION IN A SINGLE SESSION FOR THE QUERY "SUN"

SEARCH RESULTS	CLICK SEQUENCE
http://www.sunnetwork.in/	0
http://en.wikipedia.org/wiki/Sun	1
http://www.sunpictures.in/	2
http://en.wikipedia.org/wiki/Sun_Microsystems	0
http://solarsystem.nasa.gov/planets/	1
http://www.thesun.it/	1
www.thesunmagazine.org/	0
www.nasa.gov/worldbook/sun_worldbook.html	0

The Table 1 represents a feedback session and a single session for the query "sun".0 in click sequence means unclicked and 1 means clicked. All the eight URLs construct a single session. The URLs in the rectangular box construct a feedback session. Therefore for inferring user search goals, it is more efficient to analyze the feedback sessions than to analyze the search results or clicked URLs directly.

E. Inferring User Search Goals By Clustering Pseudo-Documents

1) *Clustering using bisecting K-means*: Each feedback session is represented 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

$$\text{Sim}_{i,j} = \frac{(\text{ Ffsi.Ffsj })}{(|\text{ Ffsi }||\text{ Ffsj}||)} \qquad (3)$$

The pseudo-documents are clustered using bisecting k-means clustering algorithm. After clustering all the pseudo-documents, each cluster can be considered as one user search goal. The exact number of user search goals for each query is not known, we set k to be five different values (i.e 1,2,...5). After clustering all the pseudo- documents each cluster can be considered as one user search goal.

2) *Clustering using Suffix Tree Document Clustering:* The suffix tree document clustering is organized as follows

a) *Document Cleaning:* In the document "cleaning" step, the string of text representing each document is transformed using a light porter stemming algorithm. Sentence boundaries are marked and non-word tokens (such as numbers, HTML tags, and most punctuation) are stripped.

b) *Constructing generalized suffix tree:* In the process of constructing generalized suffix tree the first step is construction of suffix tree. The suffix tree is a data structure which contains all the suffixes of a given string, so as to run many important string operations more efficiently. The string may be a string of characters or string of words. The suffix tree for the string S is defined as a tree such that: the paths from the root to the leaves have a one-to-one relationship with the suffixes of S, all edges are labeled with non-empty strings, all internal nodes (except perhaps the root) have at least two children.

Preprocessed documents

Doc 1: Jaguar Kid Planet Defender animal Wildlife

Doc 2: Jaguar animal variety tiger

Doc 3: Jaguar Car Wallpaper latest 2015

Doc 4: Jaguar Car India Price Review Photos CarWale BikeWale Offers Free Gifts

Doc 5: Jaguar Land Rover built two British car brand

Doc 6: Jaguar XF Price India Photos Review CarWale BikeWale Offers Free Gifts

Generalized suffix tree

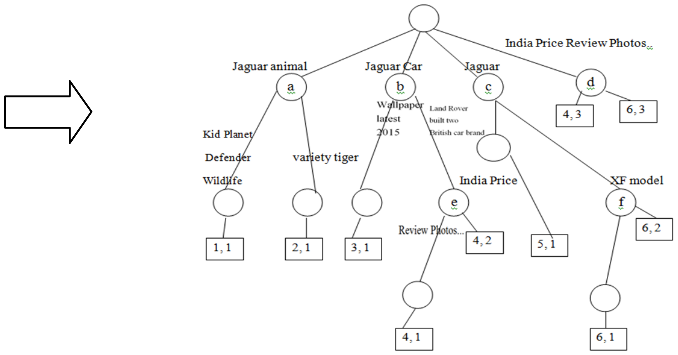


Fig. 3 Construction of Generalized Suffix Tree from preprocessed documents.

In this generalized suffix tree figure, the circle represents node, the numbers in the square represent document group. Each node of the suffix tree represents a group of documents and a phrase that is common to all of them. Once the suffix tree has been constructed, each node on the suffix tree that contain multiple documents. To reduce the number of clusters, we next undergo a cluster mergin phase.

3) *Identifying Base Clusters:* In the generalized suffix tree each node represents a base cluster. To each base cluster is assigned a score s(B) that is a function of the number of documents it contains, and the words that make up its phrase. The function is given as

$$s(B)=|B| \bullet f\,(|P|)$$

Here |B| indicates the number of document in a base cluster and |P| is the number of word in a phrase. In the table (Table 2) we can see six nodes from the example shown in figure(Figure 3)and their corresponding base clusters.

4) *Combining base clusters into clusters:* In this step, documents may be sharing more than one phrase. To avoid the document overlapping and a nearly identical cluster, this step is assigned to merge base cluster with high overlap in the document set. They defined a binary similarity measure to calculate whether base clusters should be merged or not.

$$= \begin{cases} 1 & \text{iff} \quad | B_m \cap B_n| / | B_m | > \alpha \\ & \text{and } | B_m \cap B_n| / | B_n | > \alpha \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Where α is a constant between 0 and 1. $| B_m \cap B_n|$ refers to the number of documents common to both phrase cluster and $| B_m |$, $| B_n |$refers to the number of documents in B_m and B_n. This step is presented on the figure (Figure 4). Each cluster consists of the union of the document of all its base clusters. This figure explains the base cluster graph of the six base clusters.

Table 2Representation of nodes and corresponding clusters

NODE	PHRASE	DOCUMENTS
a	Jaguar animal	1,2
b	Jaguar car	3,4
c	Jaguar	5,6
d	Indian price photo review..	4,6
e	Jaguar car Indian Price	4
f	Jaguar XF model	6

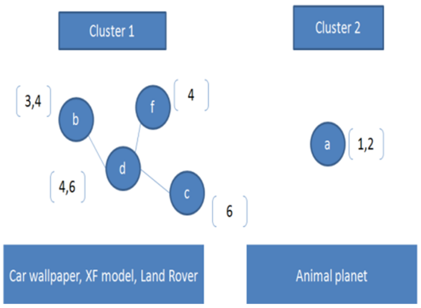


Figure 3 Result of combining base clusters

IV. EVALUATION METRICS AND RESULT ANALYSIS

A. Evaluation Metrics for Effectiveness of the Cluster

The optimal number of clusters is still not determined when inferring user search goals. The feedback information is needed to finally determine the best cluster number. In this section, we propose this novel criterion "Classified Average Precision" to evaluate the restructure results. Based on the proposed criterion, we also describe the method to select the best cluster number.

A possible evaluation criterion is the average precision (AP) 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.

$$AP=\frac{1}{N^+}\sum_{r=1}^{N} rel(r)\frac{R_r}{r} \qquad (5)$$

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 R_r is the number of relevant retrieved documents of rank r or less. VAP is the voted average precision which can be used for grouping the dissimilar documents for the particular user query search. Risk is the mapping of similar and dissimilar documents for the particular user query.

$$\text{Risk} =\frac{\sum_{i,j=1(i<j)}^{m} d_{ij}}{C_m^2} \qquad (6)$$

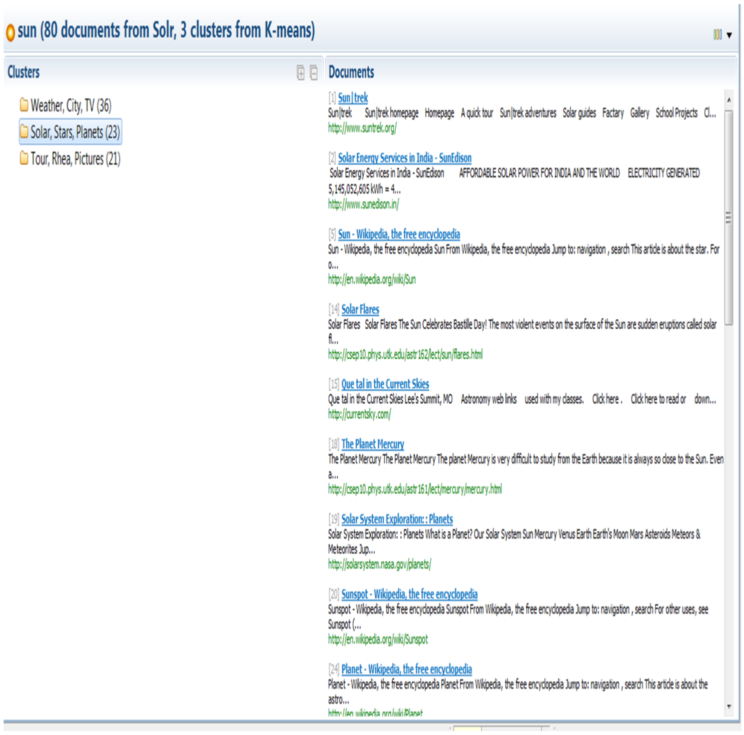
Where m is the number of the clicked URLs. If the pair of the ith clicked URL and the jth clicked URL are not categorized into one class, d_{ij} will be 1;

Otherwise, it will be 0.$C_m^2 = \frac{m(m-1)}{2}$ is the total number of the clicked URL pairs. We can further extend VAP by introducing the above Risk and propose a new criterion "Classified AP," as shown below

$$CAP=VAP \times (1\text{-}Risk)^{\gamma} \qquad (7)$$

A. Result Analysis

The dataset is generated by crawl the websites. Apache Nutch is the open source Java tool used to crawling the website. Apache Solr is the web server used to index the query term for the dataset generated by Nutch crawler. The result are analysed on various aspects such as title and snippet length, title and snippet weight, the number of the retrieved documents and depending upon the value of k. The clustered results for the query “sun” is shown in figure 2.



sun (80 documents from Solr, 3 clusters from K-means)

Clusters

- Weather, City, TV (36)
- Solar, Stars, Planets (23)
- Tour, Rhea, Pictures (21)

Documents

[1] Sun|trek
Sun|trek Sun|trek homepage Homepage A quick tour Sun|trek adventures Solar guides Factory Gallery School Projects Cl...
http://www.suntrek.org/

[2] Solar Energy Services in India - SunEdison
Solar Energy Services in India - SunEdison AFFORDABLE SOLAR POWER FOR INDIA AND THE WORLD ELECTRICITY GENERATED 5,145,052,605 kWh = 4...
http://www.sunedison.in/

[5] Sun - Wikipedia, the free encyclopedia
Sun - Wikipedia, the free encyclopedia Sun From Wikipedia, the free encyclopedia Jump to: navigation , search This article is about the star. For o...
http://en.wikipedia.org/wiki/Sun

[14] Solar Flares
Solar Flares Solar Flares The Sun Celebrates Bastile Day! The most violent events on the surface of the Sun are sudden eruptions called solar fl...
http://csep10.phys.utk.edu/astr162/lect/sun/flares.html

[15] Que tal in the Current Skies
Que tal in the Current Skies Lee's Summit, MO Astronomy web links used with my classes. Click here . Click here to read or down...
http://currentsky.com/

[18] The Planet Mercury
The Planet Mercury The Planet Mercury The planet Mercury is very difficult to study from the Earth because it is always so close to the Sun. Even a...
http://csep10.phys.utk.edu/astr161/lect/mercury/mercury.html

[19] Solar System Exploration: : Planets
Solar System Exploration: : Planets What is a Planet? Our Solar System Sun Mercury Venus Earth Earth's Moon Mars Asteroids Meteors & Meteorites Jup...
http://solarsystem.nasa.gov/planets/

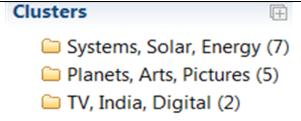
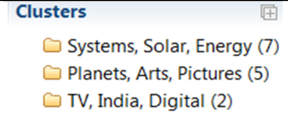
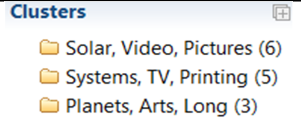
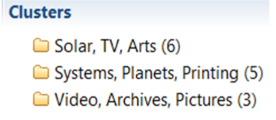
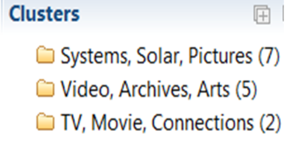
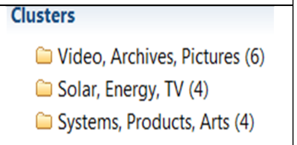
[20] Sunspot - Wikipedia, the free encyclopedia
Sunspot - Wikipedia, the free encyclopedia Sunspot From Wikipedia, the free encyclopedia Jump to: navigation , search For other uses, see Sunspot (...
http://en.wikipedia.org/wiki/Sunspot

[24] Planet - Wikipedia, the free encyclopedia
Planet - Wikipedia, the free encyclopedia Planet From Wikipedia, the free encyclopedia Jump to: navigation , search This article is about the astro...
http://en.wikipedia.org/wiki/Planet

Fig.2 Search Results for the query ‘sun’

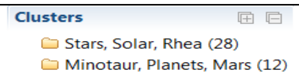
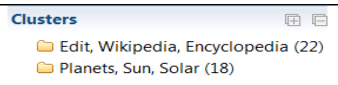
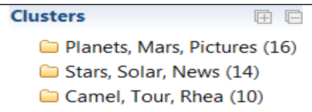
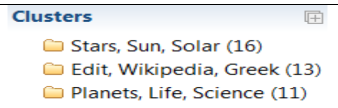
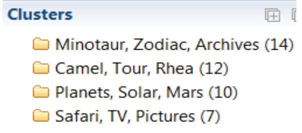
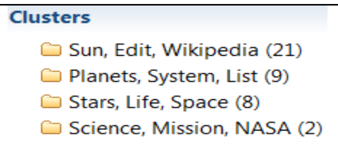
From the Table 2, by using the various values for title/snippet length, the results are similar. Although the values for length are changed it does not concern the outcome result. The k-means algorithm uses the same level of cluster contamination for both the title and snippet. The meaningful cluster labels are produced by matching the frequent phrase not only in the title but also in the snippet.

TABLE II
COMPARING THE PARAMETER VALUE OF TITLE/SNIPPET LENGTH

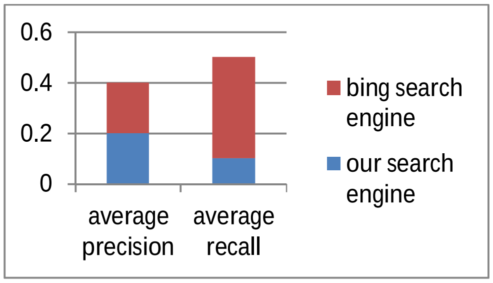
	Length of title/snippet value=100	Length of title/snippet value=200
Title weight=2 Snippet weight=1	 Clusters Systems, Solar, Energy (7) Planets, Arts, Pictures (5) TV, India, Digital (2)	 Clusters Systems, Solar, Energy (7) Planets, Arts, Pictures (5) TV, India, Digital (2)
Title weight=1.5 Snippet weight=1	 Clusters Solar, Video, Pictures (6) Systems, TV, Printing (5) Planets, Arts, Long (3)	 Clusters Solar, TV, Arts (6) Systems, Planets, Printing (5) Video, Archives, Pictures (3)
Title weight=1 Snippet weight=1	 Clusters Systems, Solar, Pictures (7) Video, Archives, Arts (5) TV, Movie, Connections (2)	 Clusters Video, Archives, Pictures (6) Solar, Energy, TV (4) Systems, Products, Arts (4)

By analysing the various values for the title and the snippet weight, the efficient result is obtained by keeping the weight of title as 2 and weight of snippet as 1 and also the irrelevant results are reduced. In order to find the correct values for k, k can assign the values from k=1,2,...5 and set title weight to be 2 and snippet weight to be 1 and it can be analysed from the following Table 3 for the various different queries.

TABLE III
COMPARISON OF K-VALUE WITH VARIOUS QUERIES

K value	Query sun	Query earth
K=2	 Clusters Stars, Solar, Rhea (28) Minotaur, Planets, Mars (12)	 Clusters Edit, Wikipedia, Encyclopedia (22) Planets, Sun, Solar (18)
K=3	 Clusters Planets, Mars, Pictures (16) Stars, Solar, News (14) Camel, Tour, Rhea (10)	 Clusters Stars, Sun, Solar (16) Edit, Wikipedia, Greek (13) Planets, Life, Science (11)
K=4	 Clusters Minotaur, Zodiac, Archives (14) Camel, Tour, Rhea (12) Planets, Solar, Mars (10) Safari, TV, Pictures (7)	 Clusters Sun, Edit, Wikipedia (21) Planets, System, List (9) Stars, Life, Space (8) Science, Mission, NASA (2)

From the Table 3, with the value of k=3, optimum results are obtained. The optimal values of k are still clearly determined using the CAP with highest value. By experimenting the various queries with CAP, we found that CAP has higher values when the value of K=3.



When we measure the precision and recall for various queries with Bing search engine and our proposed feedback search engine, we get the average precision and average recall as in the figure. For example when the queries such as “jaguar”,”apple”,”sun” ,”rain” are given to the search engine, the Bing has lower precision because it displays of highly searched results but our searched engine which cluster the results taking click through logs. Hence our search engine shows the higher precision for search results.

V. CONCLUSION

The feedback sessions are represented by pseudo-documents and are used to restructure web search results. The feedback sessions to be analyzed to infer user search goals rather search results or clicked URLs. Therefore, feedback sessions reflect user information needs more efficiently. Each URL in feedback sessions is enriched with additional textual contents including the titles and snippets represented by the pseudo-documents. Based on the pseudo-documents, user search goals can then be discovered and depicted with some keywords. The pseudo-documents are clustered using the clustering algorithm. The discovered clusters of query are used to assist users in web search.

VI.ACKNOWLEDGEMENT

First of all I would like to extend my sincere thanks to my supervisor A.P.Ponselvakumar.M.E., for providing the opportunities of taking the part in Master of Computer and Communication Program and his ideas and suggestions, which have been very helpful in the project. I am deeply thanks to Dr.C.Nalini for her valuable suggestions and many discussions.

REFERENCES

[1] U.Lee, Z.Liu and J.Cho (2005) , ‘Automatic Identification of User Goals in Web Search’, In Proc. 14th Int ’l Conference on World Wide Web (WWW ’05), Vol. 51, No. 3, pp. 391-400.

[2] R..Jones, B.Rey, O.Madani and W.Greiner (2006),‘Generating Query Substitutions’, In Proc. 15th Int ’l Conference on World Wide Web (WWW ’06),

Vol. 62, No. 6, pp. 387-396.

[3] D. Shen, J. Sun, Q. Yang, and Z. Chen(2006), "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.

[4] X.Wang and C. X Zhai(2007), 'Learn from Web Search Logs to Organize Search Results', In Proc. of 30th Annual Int 'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07), Vol. 71, No. 5, pp. 87-94.

[5] B. Poblete and B.-Y Ricardo(2008), "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.

[6] D.Beeferman and A.Berger(2000), 'Agglomerative Clustering of a Search Engine Query Log', In Proc. 6th ACM SIGKDD Int 'l Conference on Knowledge Discovery and Data Mining (SIGKDD '00), Vol. 27, No. 2, pp. 407-416.

[7] H. Chen and S. Dumais(2000), 'Bringing Order to the Web: Automatically Categorizing Search Results', In Proc. SIGCHI Conference on Human Factors in Computing Systems (SIGCHI '00), Vol. 17, No. 3, pp. 145-152.

[8] H.Cao, D.Jiang, J.Pei, Q.He, Z.Liao, E.Chen, and H.Li(2008) , 'Context-Aware Query Suggestion by Mining Click-Through', In Proc. 14th ACM SIGKDD Int 'l Conference on Knowledge Discovery and Data Mining (SIGKDD '08), Vol. 35, No. 2, pp. 875-883.

[9] X.Li, Y. Y Wang and A.Acero (2008), 'Learning Query Intent from Regularized Click Graphs', In Proc. 31th Annual Int 'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08), Vol. 43, No. 5, pp. 339-346.

[10] Zheng Lu, Hongyuan Zha, Xiaokang Yang, Weiyao Lin and Zhaohui Zheng(2013) , 'A New Algorithm for inferring User Search Goals with Feedback Sessions', IEEE Transaction on Knowledge and Data Engineering, Vol. 25, No. 3, pp. 502-522.

[11] H.-J Zeng, Q.-C He, Z. Chen, W.-Y Ma, and J. Ma, (2004)"Learning to Cluster Web Search Results,"Proc. 27th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval.