International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.1, pp-036-040 ISSN: 2249-6645

PERSONALIZATION CONCEPT BASED USER PROFILE ON SEARCH ENGINE

D. Nagesh[1]
[1]PG Scholar
Department of IT, Aurora's Engineering College,
Bhongir, Nalgonda, Andhra Pradesh, India

M. V. Vijaya Saradhi[2]
[2]Professor& HOD
Department of IT, Aurora's Engineering College,
Bhongir, Nalgonda, Andhra Pradesh, Indi

Abstract: In the present days the search engines return the same results for the same query, regardless of the user's interest. So in order to avoid that we introduce new concept called personalization. The Personalized Search aims to customize search results according to each individual user for him to find the most relevant documents to him on the top by considering his idiosyncrasies. This could possibly satisfy them and help in finding relevant information easily and quickly. User profile is a component of any personalization applications. Most existing user profiling strategies are based on objects that users like, but not the objects that users dislike. In this paper, we focus on search engine personalization and develop concept-based user profiling methods that are based on both preferences. Users can be mined from the concept-based user profiles to perform mutual filtering. Browsers with same idea and can share their knowledge

Keywords: User Profile, Personalization, Clustering, Query logs, Ranking, Search Engine.

I.INTRODUCTION

There has been a tremendous growth in the amount of information on the web. Information retrieval systems are critical for overcoming this information overload and providing the information of interest to users of the systems. Users typically pose a short query consisting of a few keywords describing their information need. Information Retrieval systems perform a 'word to word' match of the query words with all documents in their document collection and return documents containing the words entered. Retrieval in a web scenario is much harder due to the large and dynamic content on the web.

Major web search engines usually cater to hundreds of millions of users and hundreds of millions queries every day. It is very unlikely that the millions of users are similar in interests and search for similar information. Also, it is probable that the query words entered by users exhibit polysemy (same word used in different senses like 'Java' can be used to mean Java the programming language or Java islands in Indonesia) and synonymy (different words can be used to convey similar information like OOP and Object Oriented Programming) due to ambiguous nature of natural language. Therefore, given different backgrounds of users, different interests of users and ambiguities in natural language, it is very likely that query words of two different users may appear exactly same even though information needs are different. However, current retrieval systems perform a 'word to word' match of the query words and work in a "one size fits all" fashion using the same search procedure for all the users. This makes the current retrieval systems far from optimal.

This inherent non-optimality is seen clearly in the following three cases: [1] When a query contains ambiguous terms: Different users may use exactly the same query (e.g., "Java") to search for different information (e.g., the Java island in Indonesia or the Java programming language), but existing IR systems return the same results for these users. Without considering the actual user, it is impossible to know which sense "Java" refers to in a query. [2] When a query contains partial information: A query can contain an acronym or a shorter usage of a longer phrase. Then there might not be sufficient information required to infer information need of user. For example a query like "SBH" can mean "State Bank of Hyderabad" or "Syracuse Behavioral Healthcare" among others. Existing IR systems return mixture of results containing the exact word which might contain different expansions. Knowledge of interests and/or location of the user could be helpful in gathering more information required to understand the query. [3] When information need of the user changes: A users information needs may change over time. The same user may use "Java" sometimes to mean the Java island in Indonesia and some other times to mean the programming language. Without recognizing the search context, it would be again impossible to recognize the correct sense. Thus using user context information about user and query is necessary for improving the retrieval performance. Indeed, personalized search essentially boils down to capturing and exploiting related user context information of a query to improve search accuracy.

II.RELATED WORK

In the today's world because of advancement in the technology and increase in the computer literacy, computer and internet are becoming the most necessary part of human life. And people are using search engines to search necessary information. The search engine has the different type of users.

A. A brief history of web searching

Search engines as we know them today began to appear in 1994 when the number of HTTP resources increased. However, Internet search engines were in use before the emergence and growth of the Web. The first pre-Web search engine was Archie, which allowed keyword searches of a database of names of files available via FTP. The first robot and search engine of the Web was Wandex, which was developed by Matthew Gray in 1993. Since the appearance and exponential growth of the Web, hundreds of search engines with different features have appeared.

Primary search engines were designed based on traditional information retrieval methods. AltaVista, Lycos and Excite made huge centralized indices of Web pages. To answer a query, they simply retrieved results from their indexed databases and showed the cached pages based on keyword occurrence and proximity. While traditional indexing models have been successful in databases, it was revealed that these methods are not sufficient for a tremendously unstructured information resource such as the Web. The completeness of the index is not the only factor in the quality of search results. "Junk results" often wash out any results that a user is interested in. In order to increase the quality of search, Google made an innovative ranking system for the entire Web. PageRank used the citation graph of the Web and Google introduced link analysis in the search engine systems. Other efforts have been made to customize and specialize search tools.

Current retrieval systems (or search engines) return a long list of results obtained by 'word to word' match with query words. However, it has been observed that users typically view only top few (usually [10]) documents out of the long list of results returned by search engines. This requires retrieval systems to show the most relevant documents to a user on the top to improve user satisfaction with the search engine. However, without knowledge about the user context, this task is di cult to do because"relevance" of a document depends on the individual user and the individual query.

	Short term (dynamic)	Long term (static)
Explicit	immediately judged relevant document	hobbies, occupation interests
Implicit	immediately clicked document	query log

Table 1.1: Classification and examples of User Context

III.PERSONALIZATION SEARCH

A. Conceptual Based Search

Most concept-based methods automatically derive users' topical interests by exploring the contents of the users' browsed documents and search histories. Liu et al. [13] proposed a user profiling method based on users' search history and the Open Directory Project (ODP) [16]. The user profile is represented as a set of categories, and for each category, a set of keywords with weights. The categories stored in the user profiles serve as a context to disambiguate user queries. If a profile shows that a user is interested in certain categories, the search can be narrowed down by providing suggested results according to the user's preferred categories.

Gauch et al. [9] proposed a method to create user profiles from user browsed documents. User profiles are created using concepts from the top four levels of the concept hierarchy created by Magellan [14]. A classifier is employed to classify user browsed documents into concepts in the reference ontology. Xu et al. [20] proposed a scalable method which automatically builds user profiles based on users' personal documents (e.g. browsing histories and emails). The user profiles summarize users' interests into hierarchical structures. The method assumes that terms exist frequently in user's browsed documents represent topics that the user is interested in. Frequent terms are extracted from users' browsed documents to build hierarchical user profiles representing users' topical interests.

Liu et al. and Gauch et al. both use reference ontology (e.g. ODP) to develop the hierarchical user profiles, while Xu et al. automatically extracts possible topics from users' browsed documents and organizes the topics into hierarchical structures. The major advantage of dynamically building a topic hierarchy is that new topics can be easily recognized and extracted from documents and added to the topic hierarchy, whereas reference ontology such as ODP is not always up to-date. Thus, all of the proposed users profiling strategies rely on a concept extraction method, which extracts concepts from web-snippets2 to create accurate and up-to-date user profiles.

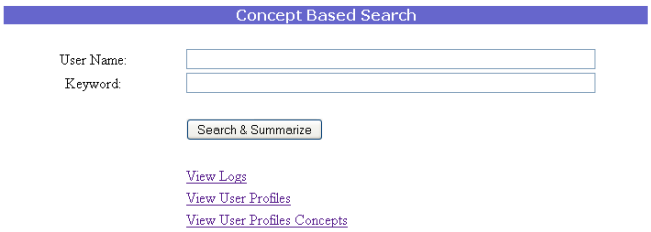

Fig1.Concept Based Search

B. Document Based Search

Most document-based methods focus on analyzing users' clicking and browsing behaviors recorded in the user's clickthrough data. On web search engines, clickthrough data is an important implicit feedback mechanism from users. Clickthrough data for the query "apple", which contains a list of ranked search results presented to the user, with identification on the results that the user has clicked on.

Joachim's [10] proposed a method which employs preference mining and machine learning to model users' clicking and browsing behavior. Joachim's method assumes that a user would scan the search result list from top to bottom. If a user has skipped a document d_i at rank i before clicking on document d_j at rank j, it is assumed that he/she must have scan the document d_i and decided to skip it. Thus, it can be concluded that the user prefers document dj more than document d_i (i.e. $d_j < r' d_i$, where r' is the user's preference order of the documents in the search result list).

C. Query logs

Search Query logs consist of logs of searches made by users of search engines. They are usually collected at the search engine server. They typically consist of : user identity (ip address or anonymous id etc), search queries, corresponding clickthroughs made by the user and click information regarding it like the click time, no of clicks made etc. Some times the query logs are also captured on the client side i.e., on the user's computers. Clickthrough data/Query logs have been the most important source for capturing user context for user modeling. There has been some work in this connection some of which are described below.

In [6], Sugiyama et. al used web browsing history in past N days for personalized search. They partition the browsing history data into three categories according to the time stamp, i.e., persistent data (before today), today data (today but before the current session) and current session data. They found that the performance of using web browsing history is competitive with that using relevance feedback. Speretta et al[11] also used users search history to construct user profiles. Several other works have made use of past queries mined from the query logs to help the current searcher. (see [[7], [8], [9], [10]]).

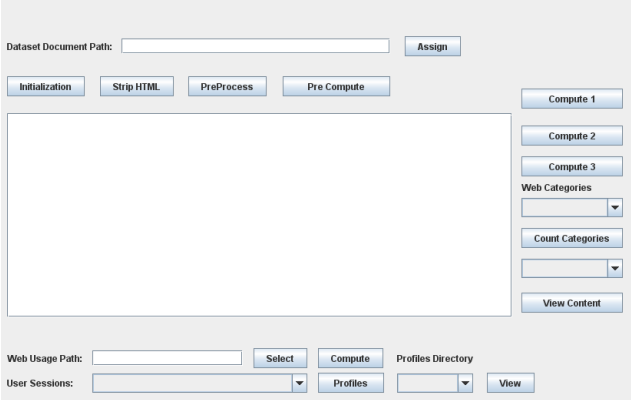

Fig2 .Concept Based Search Assigning the Databases

D. Query Clustering Algorithm

Concept-based user profiles are employed in the clustering process to achieve personalization effect.

First, a query-concept bipartite graph G is constructed by the clustering algorithm with one set of nodes corresponds to the set of users' queries, and the other corresponds to the sets of extracted concepts. Each individual query submitted by each user is treated as an individual node in the bipartite graph by labeling each query with a user identifier. Concepts with interestingness weights (defined in Equation 1) greater than zero in the user profile are linked to the query with the corresponding interestingness weight in G.

Second, a two-step personalized clustering algorithm is applied to the bipartite graph G, to obtain clusters of similar queries and similar concepts. The personalized clustering algorithm iteratively merges the most similar pair of query nodes, and then the most similar pair of concept nodes, and then merges the most similar pair of query nodes, and so on. The following cosine similarity function is employed to compute the similarity score sim(x, y) of a pair of query nodes or a pair of concept nodes. The advantages of the cosine similarity are that it can accommodate negative concept weights and produce normalized similarity values in the clustering process.

$$sim(x,y) = \frac{N_x \cdot N_y}{\|N_x\| \|N_y\|} \quad (1)$$

The algorithm is divided into two steps, initial clustering and community merging. In initial clustering, queries are grouped within the scope of each user. Community merging is then involved to group queries for the community. A more detailed example is provided in our previous work [11] to explain the purpose of the two steps in our personalized clustering algorithm

A common requirement of iterative clustering algorithms is to determine when the clustering process should stop to avoid over-merging of the clusters. Likewise, a critical issue in Algorithm 1 is to decide the termination points for initial clustering and community merging. When the termination point for initial clustering is reached, community merging kicks off; when the termination point for community merging is reached, the whole algorithm terminates.

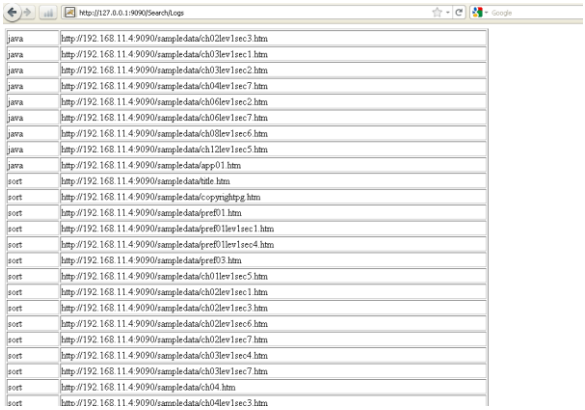

Fig 3. User Logs

Good timing to stop the two phases is important to the algorithm, since if initial clustering is stopped too early (i.e., not all clusters are well formed), community merging merges all the identical queries from different users, and thus generates a single big cluster without much personalization effect. However, if initial clustering is stopped too late, the clusters are already overly merged before community merging begins. The low precision rate thus resulted would undermine the quality of the whole clustering process

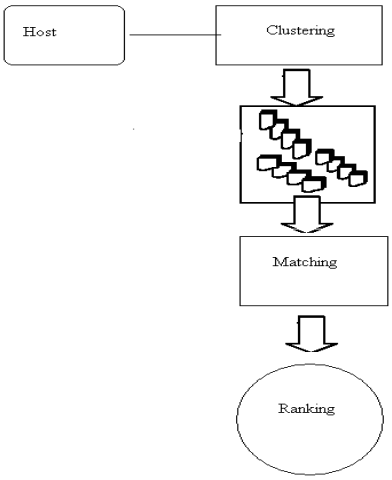

Fig 4. Document Clustering.

IV.USER PROFILING

This section proposes two user profiling strategies which are both concept-based and utilize users' positive and negative preferences.
They are
A. PJoachims−C
B. PClick+Joachims−C

A. Joachims-C Method ($P_{Joachims}$−C)

Joachims' original method was based on users' document preferences. If a user has skipped a document d_i at rank i before clicking on document d_j at rank j, he/she must have scanned the document d_i and decided to skip it. Thus, we can conclude that the user prefers document dj more than document d_i (i.e., $d_j < r' d_i$, where r' is the user's preference order of the documents in the search result list).

It is extended Joachims' method, which is a document-based method, to a concept based method (Joachims-C). Instead of obtaining the document preferences $d_j < r' d_i$, Joachims-C assumes that the user prefers the concepts $C(d_j)$ associated with document d_j to the concepts $C(d_i)$ associated with document d_i, and produces the corresponding concept preferences. From this it can be concluded that the concepts C(d5) is more relevant to the user than the concepts in the other three unclicked documents (i.e., C(d2), C(d3) and C(d4)). The concept preference pair's extracted using Joachims-C method is shown in Table 4.2.

Concept Preference Pairs for d_1	Concept Preference Pairs for d_5	Concept Preference Pairs for d_8
Empty Set	apple store $<_{r'}$ product macintosh $<_{r'}$ product	macintosh $<_{r'}$ product catalog $<_{r'}$ product
	apple store $<_{r'}$ mac os macintosh $<_{r'}$ mac os	macintosh $<_{r'}$ mac os catalog $<_{r'}$ mac os
	macintosh $<_{r'}$ apple store apple store $<_{r'}$ iPod macintosh $<_{r'}$ iPod	macintosh $<_{r'}$ apple store catalog $<_{r'}$ apple store macintosh $<_{r'}$ iPod catalog $<_{r'}$ iPod
		macintosh $<_{r'}$ fruit catalog $<_{r'}$ fruit macintosh $<_{r'}$ apple hill catalog $<_{r'}$ apple hill
		macintosh $<_{r'}$ fruit catalog $<_{r'}$ fruit

Table 4.1 Concept Preference Pairs Obtained Using Joachims-C Methods

After the concept preference pairs are identified using Proposition , a ranking SVM algorithm [10] is employed to learn the user's preferences, which is represented as a weighted concept vector. Given a set of concept preference pairs T, ranking SVM aims at finding a linear ranking function f(q, c) to rank the extracted concepts so that as many concept preference pairs in T as possible are satisfied. f(q, c) is defined as the inner product of a weight vector $\vec{w}$ and a feature vector of query-concept mapping φ(q, c), which describes how well a concept c matches the user's interest for a query q.

B. Click+Joachims-C Method (PClick+Joachims−C)

In [11], it is observed that PClick is good in capturing user's positive preferences. In this paper, it is integrated the click-based method, which captures only positive preferences, with the Joachims-C method, with which negative preferences can be obtained. It is found that Joachims-C is good in predicting users' negative preferences. Since both the user profiles PClick and PJoachims−C are represented as weighted concept vectors, the two vectors can be combined using the following formula:

W(c+j)cs=w(c)cs+w(j)cs if w(j)cs<0
W(c+j)cs=w(c)cs other wise ----------1
W(c+j)ci=w(c)ci+w(j)ci if w(j)c<0-------2

where $w(C + J)ci \in PClick+Joachims-C$, $w(C)ci \in PClick$, and $w(J)ci \in PJoachims-C$. If a concept *ci* has a negative weight in *PJoachims−C* (i.e., $w(J)ci < 0$), the negative weight will be added to $w(C)ci$ in *PClick* (i.e., $w(J)ci + w(C)ci$) forming the weighted concept vector for the hybrid profile *PClick+Joachims−C*.

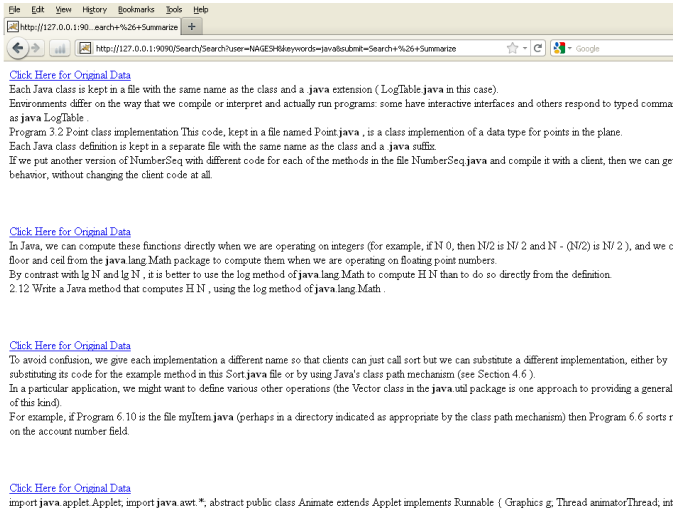

Fig4. User Personalized Based Search Results

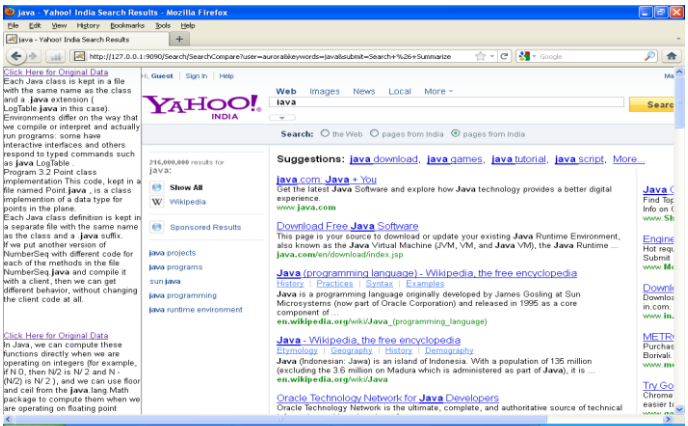

Fig5. User Personalized Comparison Search Result with yahoo

V.CONCLUSION

In this paper it has been presented a user profile strategy to improve a search engine's performance by identifying the information needs for individual users. For clustering of documents both content based clustering and session based clustering techniques is used. To automate the identification of groups of similar pages, the approach has been implemented in a Java prototype. This paper

proposes an effective method for organizing and visualizing web search results. Finally, the concept-based user profiles can be integrated into the ranking algorithms of a search engine so that search results can be ranked according to individual users' interests. To automate the identification of groups of similar pages, the approach has been implemented in a Java prototype. This paper proposes an effective method for organizing and visualizing web search results.

REFERENCES

[1]. P. Ingwersen and N. Belkin. Information retrieval in context irix. SIGIR Forum,38(2), 2004.

[2.].J. J. Rocchio. Relevance feedback in information retrieval, the smart retrieval system. Experiments in Automatic Document Processing, pages 313–323, 1971.

[3].Thorsten Joachims. Optimizing search engines using clickthrough data. In KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 133–142, New York, NY, USA, 2002. ACM
Press.

[4].D Kelly and J Teevan. Implicit feedback for inferring user preference: A bibliog-raphy. In SIGIR Forum, volume 32, 2003.

[5].M. Claypool, M. Waseda P. Le, and D. Brown. Implicit interest indicators. In Proceedings of Intelligent User Interfaces 2001, pages 33–40, 2001.

[6].K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on userprofile constructed without any e_ort from users. In Proceedings of WWW 2004,pages 675 – 684, 2004.

[7].Vijay V. Raghavan and Hayri Sever. On the reuse of past optimal queries. Pro-ceedings of the Annual International ACM SIGIR Conference on Research and
Development in Information Retrieval, pages 344–350, 1995.

[8].Jian-Yun Ji-Rong Wen and Hong-Jiang Zhang. Query clustering using user logs.ACM Transactions on Information Systems (TOIS), 20(1):59–81, 2002.

[9].Larry Fitzpatrick and Mei Dent. Automatic feedback using past queries: Social searching? In In Proceedings of the Annual International ACM SIGIR Conference
on Research and Development in Information Retrieval, pages 306–313. ACM Press, 1997.

[10].Natalie S. Glance. Community search assistant. In In Proceedings of the International Conference on Intelligent User Interfaces, pages 91–96. ACM Press,
2001.

[11].Micro Speretta and Susan Gauch. Personalizing search based on user search histories. In Thirteenth International Conference on Information and Knowledge Management (CIKM 2004), 2004.

[12].J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activites. In Proceedings of SIGIR 2005, 2005.

[13].Filip Radlinski and Thorsten Joachims. Evaluating the robustness of learning from implicit feedback. In ICML Workshop on Learning In Web Search, 2005.

[14].Boris Chidlovskii, Nathalie Glance, and Antonietta Grasso. Collaborative re-ranking of search results. In Proc. AAAI-2000 Workshop on AI for Web Search.,
2000.

[15].Henxi Lin., Gui-Rong Xue., Hua-Jun Zeng., and Yong Yu. Using probabilistic latent semantic analysis for personalized web search. In Proceedings of APWEB'05,
2005.

[16].Armin Hust. Query expansion methods for collaborative information retrieval.Inform., Forsch. Entwickl., 19(4):224–238, 2005.

Authors

D.NAGESH is currently pursuing M.Tech (WT) at Aurora's Engineering College, Bhongir, Andhra Pradesh, India.

Dr. M.V.Vijaya Saradhi received his Ph.D degree from Faculty of Engineering, Osmania University (OU), Hyderabad, Andhra Pradesh, India**.** He is Currently Working as Professor in the Department of Information Technology (IT) at Aurora's Engineering College, Bhongiri, Andhra Pradesh, India. His main research interests are Software Metrics, Distributed Systems, Object-Oriented Modeling, Data Mining, Design Patterns, Object- Oriented Design Measurements and Empirical Software Engineering. He is a life member of various Professional bodies like MIETE, MCSI, MIE, MISTE.