Improvising Web Search using Concept based Clustering

INDUMATHI.D[1], CHITRA A[2], BINEESHIA.J[3]
[1,3]Department of Computer Science and Engineering
[2]Department of Computer Applications
PSG College of Technology
Coimbatore -641004, Tamilnadu, INDIA
[1]indujaga@gmail.com, [2]ctr.psg@gmail.com, [3]bineeshiajoel.stanite777@gmail.com

Abstract: - The user profile is an elementary component of any application based on personalization. The existing strategies of user profile considers only objects which interests the user (positive preferences of the user), and not on objects which does not interest the user (negative preferences of the user). This paper focuses on personalization in search engine and the proposed approach consists of three steps. At first, an algorithm for concept extraction is employed in which concepts are extracted and the relations between these concepts are obtained from the web-snippets returned by the search engine. Second, a user profile strategy is employed to build a concept-based user profile which predicts the conceptual preferences of the user. Building user profile comprises of identifying the concept preference pair by Spy Naive Bayes Classifier (Spy NB-C) method and learning the users preferences represented by feature weights vectors by Ranking-Support Vector Machine(R-SVM). Third, the concept relations together with the predicted conceptual preferences of the user, is given as input to personalized concept-based clustering algorithm to find the conceptually related queries. To cluster ambiguous queries into different clusters of queries a personalized clustered query-concept bi-partite graph is created by making use of the extracted concepts and click through data. This suggested personalized query recommendations to the individual users based on their interests. From the experimental results, it is observed that the user profile which captured both the preferences of the user increased the separation between dissimilar queries and similar queries. Improvements in F-measure and DCG score shows that the quality of query clusters resulted provided personalised results to the users.

Key-Words: - concept extraction, negative preferences, personalization, concept clusters, search engine, user profile

1 Introduction

The recent evolution of internet is marked by the wealth of information available in the web. This evolution made the internet's destined edge, more farther than today. The internet has discovered itself on the path of expansion, so did the internet usage, with the goal of making McLuhan's Global Village a reality[1]. The Internet which serves as a giant source of information has its major task to retrieve and return the relevant piece of information for the query posed by the user, from this massive collection of information resource. But, this phenomenal growth of the web, its acceptance and exploitation by people from all walks of life, direct to exciting new challenges.

The challenge of finding appropriate responses to the queries it presents, is becoming more and more imperative. While classical techniques can and are being used on the web, it is clear that revolutionary solutions are in need, to aid users to exploit such an extremely valued resource. The intensity of challenge to retrieve user's information from the web largely depends on how properly the user can raise queries to describe the information need. Most user queries which are short and ambiguous, paves more way to new challenges, making the search an increasingly difficult task.

A study made by Spink et al.[2] examined the queries from the search engine " Excite", showed that the average search query length was only 2.4 terms. Due to these short queries the actual need of the user is not precisely expressed. For example, a user who is interested in playing cricket may use the query "cricket" to find information about tips and tricks to play cricket, whereas an entomologist (a zoologist who focuses on insects) may use the same query "cricket" to find information about the insect cricket. As a result of these short and ambiguous queries, vast amount of irrelevant pages are only retrieved. This problem can be easily shunned away if users reformulate their search query with more terms. But, they consider it as an additional burden

on their shoulders as more manual effort is involved.

Thus, the majority of the search engines offer query suggestions to the users aiding them to formulate better queries, which advances the search experience of the user. When a query is submitted by the user, a set of terms which are related semantically to the submitted query are offered back to the user, aiding them to identify what they really want, which lead to the improvement in retrieving relevant results. Unfortunately, these search engines offer the same query suggestions to the same query irrespective of the specific interest of the user. The only solution is personalization. For personalization in search engines, a user profile must be created to capture the personal preferences of the user. This is created in order to figure out the main intent of the query entered as an input and to increase the relevance of the results searched for.

The recent research focuses more on automatically learning the preference of the user. This learning is gained from documents browsed or from histories searched, which helps to build the personalized system. In contrast to the existing personalization methods that consider only the positive preference of the user, an effective personalization method which captures both the preferences of the user (positive preference and negative preference) is proposed to improve the personalized query suggestions. The elementary component in personalization of search engine is a good strategy for user profile. Few problems were noted by studying various existing strategies of user profile.

The observed problems are as follows.

- Most of the methods followed in personalization focuses mainly on creating a single user profile and apply the very same profile to every other query entered by the user as an input. The existing strategies employs only one large user profile for every user involved in the process of personalization [3, 4]. But, any query entered by the user must be looked upon differently, as the intent of the query may vary based on user preferences. For example, a user may intent to get the information regarding sports for the query “table tennis” but intent to get the information regarding insect for the query “cricket”. The existing strategies making use of only one single profile returns only information regarding sports for the query “cricket”, even though the user intents to get information regarding insect for the query “cricket”. This results in decreasing the relevancy of search results.
- The existing strategies of user profile based on clickthrough’s are categorized into concept based approach and document based approach. Both of the approaches run on the assumption that the inference of users interest can be obtained from the user clicks, although the methods used to infer the users interest, and the outcome obtained from this inference differ. The profiling methods based on documents estimate only the document preferences of the user (i.e., for some documents the preference of the user is more when compared to others).Whereas the profiling method based on concepts target to derive the concepts in which the user shows interest. While there are existence of methods based on documents, which considers what the user like and dislike, there are no other methods based on concepts which consider both the preferences to derive the conceptual interests of the user.

The existing strategies of the user profile consider only the positive preference of the user. But stepping into reality, in order to understand even the fine grain interests of the user, capturing only one side of the user preference (positive preference) may not be sufficient. Additionally another side of the user preference is required (negative preference).

For example, if a user is interested in “cricket” as a sport, and if the user has no interest in playing cricket, but enjoys watching it, he/she may be more interested in live streaming, live scores, cricket news specifically. And less interested in information regarding cricket coaching, tips and tricks in playing cricket, while absolutely no interest about the information regarding cricket as an insect. A good user profile must show favor towards cricket news, slightly favor towards cricket coaching and downgrade the information regarding cricket as an insect.

The fine detail of the user can be obtained only when the user profile is built upon both the preferences (positive and negative preferences).The strategies in personalization include even the negative preference of the user in the process of personalization, but still they are all based on documents, which does not reflect the general conceptual interests of the user.

2 Related Work

The strategies for user profile can be categorized broadly into two 1) concept based approach and 2) document based approach. The user profile methods based on documents analyze the user search queries and clicks or browsing activities of users recorded in

Table 1. An Example Clickthrough data for the query 'cricket

Doc	Clicked/ Unclicked Doc	Extracted Concepts	Doc	Clicked/ Unclicked Doc	Extracted Concepts
d1	√	Cricket News	**d5**	√	Live Cricket Score, Live Cricket Streaming
d2		Cricket Players	d6		Tips and tricks
d3		Gryllidae	d7		Cricket Insect Sound
d4		Cricket Coaching	**d8**	√	Cricket Matches, Cricket News

clickthrough data of a user. From the clickthrough data the document preferences are extracted, and then these preferences are used to learn the behaviour model of the user. The behaviour model of the user is represented by a set of feature weights. Whereas, the user profile methods based on concepts, capture the conceptual need of the user. The search histories and the browsed documents of the users are mapped automatically to a set of topical categories. The creation of user profile depends largely on the preferences of users on the extracted topical categories.

2.1 Document-Based Methods

Most of the methods based on documents analyze the user queries and browsing activities of users recorded in the clickthrough data of user. An important mechanism used for implicit feedback is clickthrough data from the users. For example Table 1 is a clickthrough data for the query "cricket". It contains a list of documents having identification on the documents in which the user has clicked on. The documents which are bolded (d1, d5, d8) are the clicked documents by the user.

The method pioneered by Joachims [3] employed machine learning and preference mining to model the user clicks and browsing activities of the user. This method assumes that the list containing search results will be scanned from top to bottom by the user. If a document d_i which is at rank i is skipped by the user and he/she clicks on a document d_j which is at rank j, then the user must have decided not to click on d_i after scanning d_i. From this it can be concluded that the document d_j is preferred more than the document d_i by the user (i.e., $d_j <_r d_i$, where r is the preference order of the user for the documents in the list containing search results). Table 2 illustrates an example set of preference pair of the documents using example clickthrough data in Table 1 and Joachims' proposition. After obtaining the preference pairs, the user preference model is learnt. This is done by employing Ranking SVM (RSVM). The user model is represented as a set of feature weights.

Table 2. An Example Preference Pairs of documents obtained using Joachims' Method

Preference pairs containing d1	Preference Pairs containing d5	Preference Pairs containing d8
Empty Set	$d5 <_r d2$	$d8 <_r d2$
	$d5 <_r d3$	$d8 <_r d3$
	$d5 <_r d4$	$d8 <_r d4$
		$d8 <_r d6$
		$d8 <_r d7$

Ng et al.[4] proposed an algorithm that combines a novel voting procedure with a spying technique to determine the document preferences of the users, from the clickthrough data. They also used a Ranking-SVM algorithm to learn the user preferences which is represented as a set of feature weights.

Some document based methods analyse user search queries for providing personalized results. Web search queries are usually short and precise. Wen et al.[5] proposed a method to cluster queries if they contain similar terms. If these terms show the way for the same set of documents, then they can be clustered together. But this method is not suitable for word sense disambiguation. So in order to identify the precise semantics of the user queries concept-based methods are needed.

2.2 Concept-Based Methods

The user profile methods based on concepts capture the conceptual needs of the user. The search histories and the browsed documents of the users are mapped automatically to a set of topical categories. The creation of user profile depends largely on the preferences of users on the extracted topical categories.

Xu et al.[6] proposed a method in which the user profiles are created automatically based on browsing histories and emails of users' (i.e. personal documents.). The users' interests are summarized into hierarchical structures. From the browsed documents of the users' the frequent terms are extracted. This method runs on the assumption that the terms which exist frequently in the browsed documents of the user's represent the users' interested topics. This is used in building hierarchical user profiles that represents the topical interests of the users.

Leung et al. [7] proposed a method to create a user concept preference profile by considering only the positive preferences of the user. In this method the concepts are first extracted for a query. The space covered by these concepts cover more concepts than the actual need of the user. To reflect the users interestingness on the concepts found in the clicked snippets, the weights of the concepts appearing in the clicked snippet are incremented by 1.The other concepts in the concept space which are related to the clicked concepts are incremented based on a similarity score. So the concepts that closely relate to the concepts clicked (neighborhood concepts) are incremented to a value close to 1 or 0.The unrelated concepts are assigned weights close to zero. Based on the interestingness on the concepts a user concept preference profile is built.

Stamou et al. [8] attempted to determine personal preferences from the users click history. For estimating the topical preferences of the user based on past searches (i.e. queries issued previously and the pages clicked for those queries) they leveraged a topical ontology. Then the semantic similarity between the current query of the user and the query-matching pages are explored inorder to find the current topic preference of the user. They have also developed a ranking function to rank the search results inorder to match the preferences of a user in a better way.

Zeng et al. [9] proposed two different approaches. The approaches are used in an e-learning system for acquisition of knowledge requirements of the user's, about the course content. The course ontology is represented as a concept hierarchy. The first approach relies on the historical session logs and interactive question-answer session. They are analyzed to determine the requirements of the user's. The second approach is based on the reading behavior logs of the users.

Bhowmick et al. [10] proposed a method to build user interest model. In this work the user interest model is constructed based on the knowledge of the domain. The knowledge of education domain is considered here. The representation of knowledge ontology, database, is organized as a three level hierarchy. The top level is the topic level that shares a parent child relationship. Second is the concept level and the third is the keyword level.

A hybrid intelligent approach has been proposed Shutan et al. [11] to automate the clustering process based on the characteristics of each document represented by the fuzzy concept networks. Through this approach, the useful knowledge can be clustered and then utilized effectively and efficiently

For personalized search, Kim et al. [12] presented a novel way to build a user profile of concept network. In this method the formal concept analysis (FCA) theory represents each concept. A session interest concept is generated, whenever a user submits a query. Then, the current concept network (i.e., a user profile which accumulates recent preferences of the user) is merged into new concepts According to FCA, the session interest concept is a pair of intent and extent where the intent consists of a set of extracted keyword features from the documents selected and the extent consists of a set of user selected documents among the search results.

Leung et al. [13] proposed a framework which supports in mining the conceptual preferences of a user from the clickthrough data of the user, resulting from Web search. To accustom a ranking function of a search engine, the preferences which are discovered are made use of. An extended set of user's conceptual preferences is derived in this framework; the preferences are based on clickthrough data and the extracted concepts from the search results. Then, the user profile is represented as a concept ontology tree by the concept-based user profile (CUP). Finally, to re-rank the search results the CUP is given as an input to a support vector machine (SVM) to learn the concept preferences of the user.

Prabaharan et al. [14] proposed an approach based on topic ontology for the generation of concept based user profile from search engine logs. The relevance of search engine results is optimized by the use of spreading activation algorithm.

Beeferman et al. [15] proposed an agglomerative clustering algorithm (BB's algorithm) to exploit query-document relationships from click through data. This algorithm is used to cluster different queries from different users. They do not provide provision for the queries that are grouped incorrectly at the early stage of clustering. This can be overcome by the Genetic algorithm based clustering approach. Personalized effect is achieved by manipulating the

user concept preference profile in the clustering process [16]

The advantage of building a topic hierarchy dynamically is that new topics can be easily extracted from the documents and added to the topic hierarchy, while using reference ontology like Open Directory Project (ODP) is not up-to-date always. Thus, our proposed user profile strategy relies on a concept extraction method, which extracts concepts from the returned web-snippets to create a user profile that is accurate and up-to-date.

3 Personalized Concept-Based Query Clustering

The approach consists of three steps. At first, an algorithm for concept extraction is employed in which concepts are extracted and the relations between these concepts are obtained from the web-snippets returned by the search engine. Second, a concept-based user profile strategy is employed to build a concept-based user profile which predicts the conceptual preference of the user. Third, the concept relations together with the predicted conceptual preferences of the user, is given as an input to a personalized concept-based clustering algorithm to find the conceptually related queries. Finally, for search refinement similar queries are obtained, which will fetch the personalized results and return those results back to the user. Figure 1 shows the general process of the proposed work.

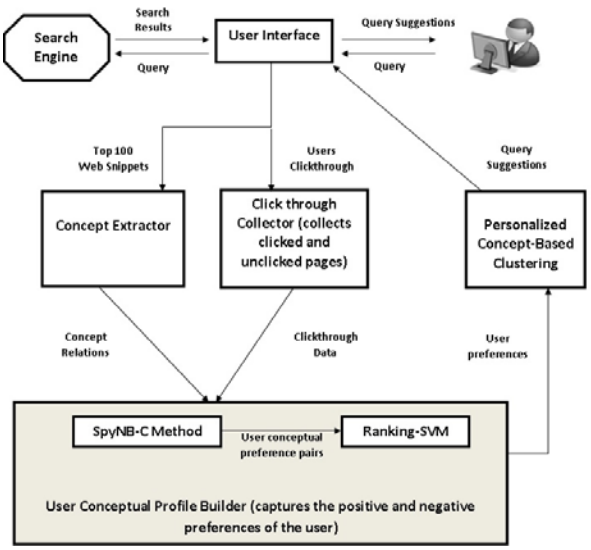

Fig. 1. The overall process of the proposed system

3.1 Concept Extraction

The concept extraction method composed of two basic steps. 1) to extract the concepts making use of the web-snippets returned back from the search engine for a user query 2) to obtain the relation between the extracted concepts.

When a user submits a query to the search engine, it returns a set of web-snippets back to the user to identify the items that are relevant to the user. If the returned web-snippets, for a particular query contain keywords or phrases that appear frequently, then those keywords or phrases will be considered as an important concept. The extracted concepts will relate to the user query, as it lies in close proximity with the user query among the top documents returned. In order to measure the interestingness of a keyword or a phrase c_i, which is extracted from the returned web-snippets of a particular query q, a support formula(Supp) is employed as follows:

$$support(c_i) = (sf(c_i)/n).|c_i| \quad (1)$$

where $sf(c_i)$ is the number of returned web-snippets containing c_i (i.e. the snippet frequency for a particular keyword or a phrase c_i), n is the total number of returned Web-snippets for a query q, $|c_i|$ is the total number of terms of a particular keyword or a phrase c_i. When a user query is submitted a set of keywords or phrases is extracted first, from the returned web-snippets, in order to extract the concepts for a particular query q. After this, the support value for all the keyword / phrase is computed. Then a threshold value s is set.

If the support value of a particular keyword or a phrase c_i is greater than the threshold value then that keyword/phrase will be considered as a concept (s=0.02 is taken in this experiment). Table 3 illustrates the extracted concepts for the query q="cricket". Before performing the concept extraction, stop words like "of", "the", etc from the returned web-snippets are removed. The limit for the maximum length of the extracted concepts is seven words. Due to this meaningful concepts are extracted and the time required for computation is found to be minimal.

To obtain relations between concepts that are extracted, a signal-to-noise ratio formula is used. To obtain relations between these extracted concepts the similarities between these concepts c1 and c2 is computed. The following similarity formula is applied directly in step 1 as follows:

$$\mathrm{sim}(t_1, t_2) = \log \frac{n.df(t_1 \cup t_2)}{df(t_1).df(t_2)} / \log n \quad (2)$$

where n is the total number of documents, df ($t_1 \cup t_2$) is the joint frequency of t_1 and t_2 in the document and df(t) is the frequency of the term t in the document. The similarity value lies between [0, 1] always, when the above formula is used.

Making use of the concepts extracted and the relations between these concepts, a concept relationship is created. Figure 2, shows an example concept relationship for the query "cricket". A relation is created between two concepts c1 and c2 if the similarity value is greater than zero and less than one.

Table 3. Extracted concepts for the query *q*="cricket" with threshold=0.02

Concept	Supp	Concept	Supp
cricket News	0.1	live streaming	0.07
cricket players	0.1	games	0.06
live cricket score	0.09	cricket insect sound	0.05
Gryllidae	0.09	cricket types	0.04
cricket buzz	0.09	cricket information	0.03
cricket videos	0.08	cricket schedules	0.03
Stridulation	0.07	cricket Insect facts	0.03
cricket coaching	0.07	cricket tips and tricks	0.03

4 Concept-Based User Profile

The space covered by the extracted concepts, cover more concepts than the actual need of the user. For example, the space covered by the concepts extracted from the web-snippets, when a user enters the query "cricket" includes "cricket live streaming", "cricket live score", "cricket news", "cricket coaching", "Gryllidae"(a family of cricket insect), "insect sound" etc. If the user shows interest towards the concept "cricket news" and selects the snippets containing the concepts "cricket news", the clickthrough data will gradually show favor towards the concept "cricket news" and its neighborhood concepts "cricket live streaming" and "cricket live score". This is done by the assignment of higher weights to the concepts" cricket news" and the weight remains zero for the unrelated concepts such as "Gryllidae" and "insect sound" Since only the positive preferences of the user is considered to build the user profile, the fine details of the users cannot be captured [7].

The document based Joachim's method is extended to a concept-based method(Joachim's-C).Instead of obtaining document preferences, concept preferences associated with the document are considered. Table 4 illustrates the concept preference pair for Joachim's-C method obtained based on Table 1.

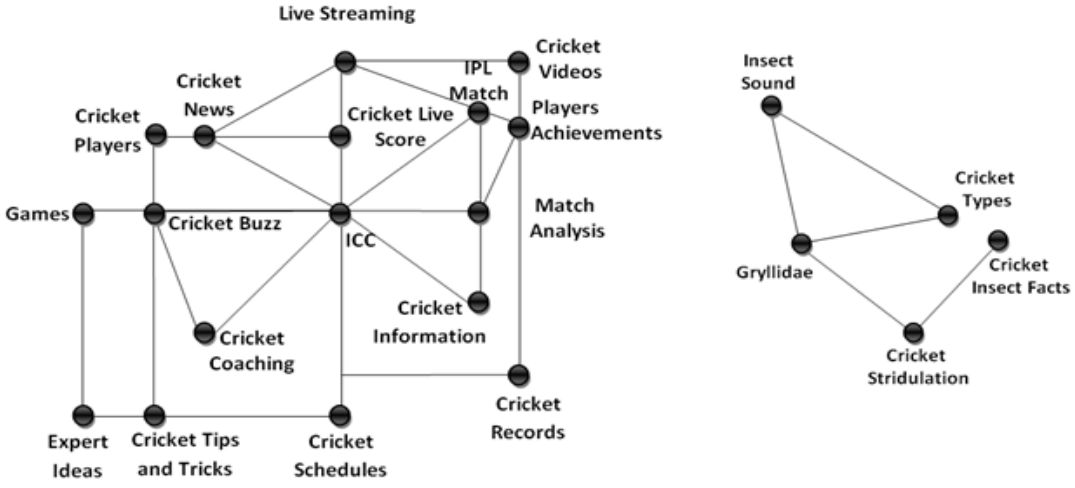

Fig. 2. An example concept relations/concept space derived for the query "cricket" without incorporating user clickthrough's

Table 4. Example Similarities between concept "players" and the rest of the extracted concepts

Concept Preference Pairs for d_1	Concept Preference Pairs for d_5	Concept Preference Pairs for d_8
Cricket News $<_r$ Cricket Players	Live Cricket Score $<_r$ Cricket Players	Cricket Matches $<_r$ Cricket Players
	Live Cricket Streaming $<_r$ Cricket Players	Cricket News $<_r$ Cricket Players
Cricket News $<_r$ Gryllidae	Live Cricket Score $<_r$ Gryllidae	Cricket Matches $<_r$ Gryllidae
	Live Cricket Streaming $<_r$ Gryllidae	Cricket News $<_r$ Gryllidae
Cricket News $<_r$ Cricket Coaching	Live Cricket Score $<_r$ Cricket Coaching	Cricket Matches $<_r$ Cricket Coaching
	Live Cricket Streaming $<_r$ Cricket Coaching	Cricket News $<_r$ Cricket Coaching
		Cricket Matches $<_r$ Tips and tricks
		Cricket News $<_r$ Tips and tricks
		Cricket Matches $<_r$ Cricket Insect Sound
		Cricket News $<_r$ Cricket Insect Sound

Joachims' method assumes that a user would scan the search result list from top to bottom. In this example, the documents d1, d5, d8 are clicked. If a user has skipped a document d2 at rank 2 before clicking on document d5 at rank 5, it is assumed that he/she must have scanned the document d2 before deciding to skip it. Thus, it can be concluded that the user prefers document d5 more than document d2. The Joachim's-C method does not capture more accurate negative samples (topics not preferred by the user).Only with accurate negative samples, reliable negative concepts can be determined. Due to this they perform worse when compared to the method used in [3].

The proposed strategy for user profile comprises of two steps: 1) to identify the concept preference pair by SpyNB-C method 2) to learn the users preferences represented by feature weights vectors by Ranking-SVM.

4.1 SpyNB-C Method

Unlike Joachim's method, SpyNB makes an assumption that the pages not clicked by the user may be either irrelevant or relevant to the user. Therefore, in SpyNB the clicked pages are treated as positive samples and the unclicked pages are treated as unlabeled samples, during the training process. The problem in finding user preferences underlies in deriving negative samples (irrelevant to the user) from the unlabeled samples. To derive the negative samples from the unlabeled samples a novel voting procedure is incorporated into a Naive Bayes classifier in the "Spy" technique. The positive classes and negative classes are denoted by "+" and "-", and the set of N documents in the list of search results are denoted by $D=d_1, d_2, ...,d_n$. For every search result in the list, the SpyNB extracts the words that appears in the URL, abstract and title.

Then a word vector ($w_1, w_2, ...,w_M$) is created for every extracted words. By estimating the prior probabilities of positive and negative samples Pr (+) and Pr (-) and the prior probabilities of the weight vector of positive and negative samples Pr($w_j|+$) and Pr($w_j|-$),a Naive Bayes classifier is built. The data used for training contains only positive samples and unlabeled samples, without considering the negative samples.

In order to learn the Naive Bayes classifier a "Spy" technique is employed. From the entire set of positive samples P, only a small set of positive samples S is selected, and is moved into U containing a set of unlabeled samples. These are called as "spies", which are used to train the Naive Bayes classifier. From the resulted classifier the probability Pr(+| d) for each sample in $U \cup S$ is assigned. If the probability assigned for each sample is lesser than the threshold Ts then an unlabeled sample is chosen as a predicted negative sample (PN). Unfortunately, most of the users would click on only few documents that are relevant to them (i.e. positive samples).Hence, only a minimum number of positive samples involves in the classification process. This lowers the probability of predicted negative samples (PN).

The above problem is resolved by making use of the entire positive samples p_i in P, to train the Naive Bayes classifier. So the entire positive samples p_i in P is chosen to move into U as "spies" and the predicted negative samples ($PN_1,PN_2, ...,PN_n$)are created using the Naive Bayes classifiers. Finally, the PN_i is combined into the final PN by using a voting procedure. SpyNB algorithm is discussed in [4]. The page preferences are generalized into concept preferences in SpyNB-C method. To obtain concept preference pairs T, the concepts $C(d_j)$ in the

positive sample d_j are considered to be more relevant than the concept $C(d_i)$ in the predicted negative sample d_j(i.e., $C(d_j) <_r C(d_i)$).

4.2 Ranking - SVM

After identifying the concept preference pairs, the user's preferences must be learnt. This is done by making use of a ranking SVM algorithm. The user's preferences are represented as a weighted concept vector. The concept preference pairs T and the feature vectors are given as an input to the Ranking-SVM algorithm. $\Phi(q,c)$ is the query-concept mapping feature vector. For a user query q, $\Phi(q,c)$ describes how well a concept c matches the interest of the user. The extracted concepts for a query q are taken and the feature vector for every concept c_i is created as follows:

$$\Phi(q,c_i)=[Feature\_c_1, Feature\_ c_2, ...,Feature\_c_n] \quad (3)$$

The feature vectors can be defined as: 1) if k = i, then $Feature\_c_k$ =1, 2) if $simR(c_i,c_k)$ >0 then $Feature\_c_k = simR(c_i,c_k)$ 3) otherwise, $Feature\_c_k$ =0.By taking T and $\Phi(q,c_i)$ as an input to Ranking SVM , weight vector $\vec{w}$ is got as output. The weight vector $\vec{w}$ must hold the maximum with the following inequalities:

$$\forall(c_i,c_j)\in r_k,(1\leq k \leq n) : w.\ \Phi(q_k,c_i)> w.\Phi(q_k,c_j) \quad (4)$$

$(c_i <_{r_k} c_j)$ is the concept preference profile of a query q_k, which means in the target concept ordering of r_k ,c_i ranks higher than c_j. $(c_i,c_j)\in r$ is a concept preference which corresponds to $(c_i <_{r_k} c_j)$. The weight vector $\vec{w}$ $(w_{Feature\text{-}c1},w_{Feature\text{-}c2}, ...,w_{Feature\text{-}cn})$ which is obtained as the output of the Ranking-SVM algorithm, determines the preference of the user on the concepts extracted. Therefore, for every extracted concepts c_1, c_2, ...,c_i, for a query q, the preference of the user are represented as the weight value $w_{Feature\text{-}c1}$, $w_{Feature\text{-}c2,}$ ...,$w_{Feature\text{-}cn}$. This creates a conceptual preference profile for a query q.

$$P_{SpyNB\text{-}C}=(w_{Feature\text{-}c1},w_{Feature\ \text{-}c2}, ...,w_{Feature\text{-}cn}) \quad (5)$$

Table 5 shows an example feature weights, which is resulted from the Ranking-SVM training for the query q="cricket" (the topical preference are "News","Streaming","Games") using SpyNB-C method from the experiment.

Table 5. Example feature weights resulted from RSVM Training for the query "cricket"

Feature	Weight	Feature	Weight
News	1.98	Streaming	1.84
players	0.563	Games	0.762
score	2.42	Sound	-0.036
Gryllidae	-0.097	Types	-0.074
Buzz	1.295	Information	-0.896
Videos	-0.341	Schedules	1.027
Stridulation	-0.092	Facts	-0.198
Coaching	0.132	Tips	0.241

5 Concept-Based Clustering

To cluster ambiguous queries into different clusters of queries the idea of personalized concept-based clustering is adopted. The approach used for personalized concept based clustering consists of two steps: 1) to construct a query-concept bi-partite graph by making use of the extracted concepts and the clickthrough data 2) to construct a personalized clustered query-concept bi-partite graph making use of the query-concept bi-partite graph obtained in step 1. To achieve personalization effect concept-based user profiles are employed in the process of clustering.

5.1 Bipartite Graph Construction

By making use of the algorithm for personalized concept-based clustering, a query-concept bipartite graph G is constructed .The graph consists of two sets of vertices: 1) One set of vertices corresponds to a set of queries entered by the user and 2) another set of vertices corresponds to a set of concepts extracted. In the bipartite graph, each individual query submitted by each user is considered as an individual vertex. Each individual query is labeled by a user identifier. If the interestingness weights of the concepts in the user profile are greater than zero, they are linked to the query which corresponds to the interestingness weight in G.

For example, the query "cricket" submitted by two users user1 and user3 becomes two vertices "cricket(user1)" and "cricket(user3)".As per the data recorded in the user profile, if user1 shows interest towards the concept "cricket live streaming" a link between the query "cricket(user1)" and the concept "cricket live streaming" would be created. And if user3, shows interest towards the concept "cricket insect sound", a link between the query

"cricket(user3)" and the concept "cricket insect sound" would be created. Algorithm 1 shows the construction of bipartite graph.

Algorithm 1 Constructing Bipartite Graph
Input: Extracted Concepts E, Clickthrough data CT
Output: Query-Concept Bipartite Graph G
1: Obtain a set of queries entered by the user Q = {q1, q2, q3 ...} from CT
2: Obtain a set of concepts extracted C = {c1, c2, c3 ...} from E
3: Nodes (G) = Q U C where Q and C are the two sides in G
4: If the interestingness weights of the concepts in the user profile are greater than zero, they are linked to the query which corresponds to the interestingness weight in G, so an edge e=(q_i,c_j) is created

5.2 Personalized Clustering

After the construction of concept-based bi-partite graph G a two-step personalized clustering algorithm on the concept-based bipartite graph G is applied, to obtain similar query clusters and similar concept clusters. The personalized clustering algorithm is detailed in [7]. The algorithm merges the similar pair of query nodes first, and then, the similar pair of concept nodes. This is done iteratively. Computation of the similarity score sim(x,y) for a query node pair or a concept node pair is employed using cosine similarity function . Since, negative concept weights are produced cosine similarity is used to accommodate them. In the clustering process, cosine similarity produces normalized similarity values. The cosine similarity formula is as follows:

$$sim(x,y) = \frac{N_x N_y}{\| \mathrm{Nx} \| . \| \mathrm{Ny} \|} \quad (6)$$

where N_x and N_y is the weight vector in the bipartite graph G for a set of neighbor nodes of node x and node y respectively, and the weight of a neighbor node n_x in N_x and n_y in N_y in G is the link weight connecting n_x and x and n_y and y respectively.

The algorithm consists of two steps: 1) initial clustering and 2) community merging. To group queries of each user the initial clustering is employed, to group queries for the community the community merging is employed. The reason for two steps in the personalized clustering algorithm is to make sure not to lose the effect of personalization. There are higher chances for the generated query clusters to merge together with the queries entered by different users.

For example, concept nodes such as "information" may be common to cricket (user1) and cricket (user3).Both the users connect to the concept node "information". So there are higher chances for these two query nodes to be merged together in the first few iterations itself, which will cause many other queries given by different users to be merged in the succeeding iterations. By considering the example again, if cricket (user1) and cricket (user3) are merged, then in the next few iterations the concept nodes "cricket live streaming", "cricket insect sound" and "information" will get merged.

More queries across the user will be clustered together as the clustering algorithm goes further. In the end, the obtained clusters of queries will have no personalization effect. Hence to resolve this problem the two–step personalized clustering algorithm is used. So the output of the two–step personalized clustering algorithm will be optimal clusters.

As an example, cricket (user1) and cricket (user3) belongs to different clusters, which are the optimal clusters. Considering another example, to illustrate the effectiveness of the two–step personalized clustering algorithm, user1 submits the query "cricket" in order to seek information about enjoying cricket and user 2 submits the query "cricket" in order to seek information about playing cricket, while user3 submits the query "cricket" to look for information about "cricket as an insect" In this example, even though the query "cricket" submitted by user1, user2, and user3 appear to be the same, the algorithm can successfully differentiate them to archive personalization effect according to individual user conceptual preferences considering both positive and negative preferences.

Finally, the queries about enjoying cricket (e.g., "cricket live streaming", "cricket live score", "cricket news") are suggested to user1, queries about playing cricket (e.g., "cricket coaching", "tips and tricks") are suggested to user2, while queries about "cricket as an insect" (e.g., "cricket insect sound", "Stridulation", Gryllidae") are suggested to user3.

6 Experimental Results

In this section the experimental evaluations that have been performed are explained. When the user submitted a query, search engine returned search

results for the user query. Only the top 100 web-snippets were retrieved and displayed to the user. Since, most of the users would examine only the top few search results (say 10), a concept extraction method is used to dig down deep into the 100 search results.

The concept extraction method extracted the frequently occurring keywords or phrases from the web-snippets and the support value is calculated for every keyword or phrase using Equation (1). When the support value exceeded the threshold value (threshold value=0.02 in our experiment) then that keyword or phrase was considered to be an important concept. A small value for threshold is chosen in order to extract as many concepts as possible.

After the concept extraction a concept relation between these concepts were established using the similarity formula as in Equation (2) without considering the preferences of the user. A concept relation between these concepts were established, only when the similarity value lied between [0, 1]. The obtained concept relation gets stored in the database.

The clickthrough data was collected when the user clicked on the required web-snippets which were relevant to the queries. The clickthrough data consisted of concepts in the web-snippets which were clicked and those which were unclicked. Using the extracted concepts and the clickthrough data the user profile was built by SpyNB-C method which considered both the preferences of the user (positive and negative preference).

The SpyNB-C method which considered both the preferences of the user yielded the concept preference pair's. Ranking SVM was applied on the obtained concept preference pairs to learn the user's profile. The Ranking SVM computed the feature weights for the user query (i.e. the user preferences on the extracted concepts).

Then the query-concept bipartite graph was constructed making use of the clickthrough data and the extracted concepts. Then the two–step personalized query clustering algorithm was applied on the query-concept bipartite graph. At first, initial clustering was employed to group queries of each user using the similarity formula as in Equation (6) and then community merging was employed to group queries for the community using the same similarity formula. Finally a personalized query concept bipartite graph was obtained which suggested personalized query expansion to the individual users based on their interests.

The existing user profile strategy which considers only the positive preference of the user was implemented and the precision and recall value was calculated. Queries from the dataset Yahoo Webscope [17] were taken. 25 students from CSE department of PSG College of Technology were invited to search 250 test queries from the dataset. Then the students were asked to click on the web-snippets of the returned results that are both relevant to the queries and their information needs. Table 6 shows the statistics of the collected clickthrough data for this experiment.

Table 6. Statistics of the Clickthrough Data Collected

Statistics	
Number of users	25
Number of queries assigned to each user	10
Number of queries	250
Maximum number of retrieved URL's for query	100
Maximum number of extracted concepts for a query	196

The curve of precision versus recall results by averaging the results of various queries. Since the recall levels of the individual queries are not equal to standard recall levels, interpolation is used to define the precision at the standard recall levels.

The average precision versus recall of the search system which uses only positive preference user profile (Pp-Up) and the search system which uses both positive and negative preferences user profile (Pp&Np-Up) was plotted in figure 3. The user profile which captures only the positive preferences of the user yielded a worse precision and recall rating when compared to the user profile that captured both the preferences of the user.

Pp-Up captures positive preferences based on user clicks, so wrong click made by users has little effect on the final result as long as the number of wrong clicks is much less than that of correct clicks. Sometimes wrong positive predictions may significantly lower the weight of a positive concept. Thus the combination of positive and negative preferences helps to group similar queries together and dissimilar queries into different clusters. Hence Pp&Np-Up achieves better precision and recall values compared to Pp-Up

The obtained precision and recall values for the existing user profile strategy with positive preferences was compared with the precision and recall values of the proposed user profile strategy which considered the positive and negative preferences of the user. The average precision obtained for the existing system is 0.67 and the

average precision obtained for the proposed system is 0.81. It is observed that the average precision of the proposed system outperforms the average precision of the existing system by 21%. Figure 4 shows the F-measure performance comparison of the search system having user profiles Pp-Up and Pp&Np-Up. Increase in F-measure shows that it discovers the negative preferences of the user more accurately. The proposed work is a scalable one and it can perform well in high performance computing systems.

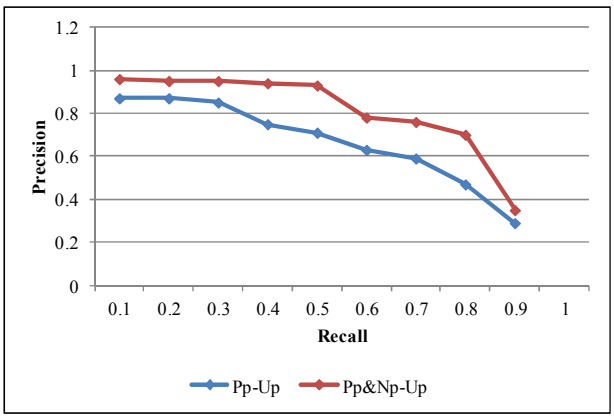

Fig. 3. Average Recall vs Precision

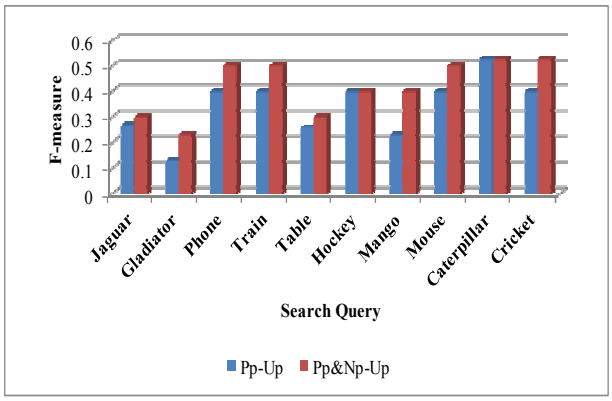

Fig.4. Performance Comparison – F measure

Ranking preferences of users were collected on a 0-5 point scale (0 –irrelevant and 5-most relevant). The ranking preferences were based on the users information need. The rating reflects the documents relevancy to the corresponding query. The ranking quality is measured using Discounted Cumulative Gains (DCG) metric. DCG is a common IR metric that represents how good a particular ordering of search results is by comparing the order of the results to the experts relevance judgment for each result [18]. For a given query q, DCG is defined as

$$DCG(q) = (2R(d) - 1)/\ln(1 + d) \quad (7)$$

where R(d) refers to relevance judgment given by experts for the document d. An ordering in which the relevant results are listed near the top will have a higher DCG. To facilitate cross query comparison, the value is normalized between 0(irrelevant) and 1(most relevant). Figure 5 illustrates the Normalized DCG Score of the search system having user profiles Pp-Up and Pp&Np-Up. A high value of DCG score for the user profile Pp&Np-Up proves that a correct user profile with positive and negative preferences can greatly improve a search engine's performance by identifying the users' information needs.

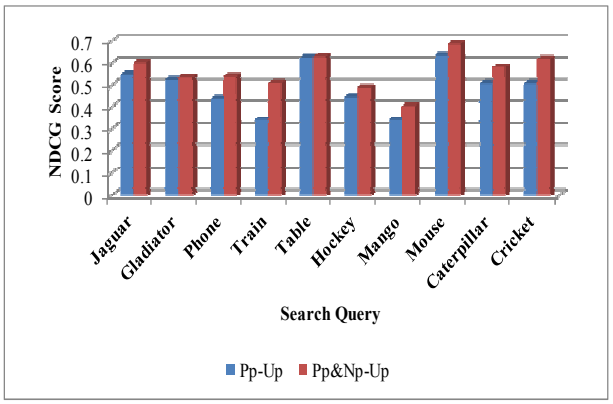

Fig. 5. Performance comparison- NDCG Score

7 Conclusion

By knowing the intended information requirement of the user implicitly, the performance of the search engine can be improved. A user profile is needed to recognize the information need of the users. In this paper a user profile strategy is proposed and evaluated. To automatically build the user profile based on concepts clickthrough data is taken and the concepts are extracted. A voting procedure is incorporated in a Naive Bayes classifier to infer both the preferences (positive and negative preferences) of the user. RSVM was employed to learn the user preferences. Experimental results with the increase in F-measure and DCG Score indicated that the user profile which captured both the preferences of the user provides personalized results to the users.

The following future work has been planned. Collaborative filtering can be performed by obtaining relationship between users. This can be

retrieved from the user profiles based on concepts. This permits the users having same interests to share their user profiles. Using the existing user profiles the intent of the unseen queries (i.e., when a user submits a new query) can be predicted, such that, even the unseen query can be benefited by personalization. This can be incorporated into a ranking algorithm of a search engine so that the search results can be ranked based on the personal interests of the user.

References:

[1] McLuhan and Marshall, *Understanding Media*. Gingko Press, 2003

[2] Spink, A., Wolfram, D. Major, B, Jansen, J., Saracevic, T, Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, Vol.52, No.3, 2001, pp.226-234.

[3] Joachims, Optimizing search engines using clickthrough data. Proc. of ACM SIGKDD Conference, 2002

[4] Ng, W., Deng, L., Lee, D L, Mining user preference using spy voting for search engine personalization. *ACM Trans. Internet Technology*, Vol.7, No.4, 2007, pp.1-27

[5] Wen, J., Nie, J., Zhang, H., Query Clustering using user logs, *ACM Trans. Information Systems*, Vol. 20, No.1, 2002, pp. 59-81.

[6] Xu, Y. Wang, K.., Zhang, B., Chen, Z., Privacy-enhancing personalized web search. *Proceedings of WWW Conference*, 2007

[7] Leung, K. W. T., Ng, W., Lee, D.L, Personalized concept-based clustering of search engine queries. *IEEE Trans. Knowledge and Data Eng.*, Vol.20, No.11, 2008, pp.1505-1518.

[8] Stamou, S., Ntoulas, A., Search personalization through query and page topical analysis. *User Modeling and User-Adapted Interaction.*, Vol.19, No.2, 2009, pp. 5-33.

[9] Zeng, Q., Zhao, Z., Liang, Y., Course ontology-based user's knowledge requirement acquisition from behaviours within e-learning systems. *Computers and Education*, Vol.53, No.3, 2009, pp. 809–818.

[10] Bhowmick, P. K., Sarkar, S., Basu, A. 2010. Ontology based user modeling for personalized information access, *Int. J. Comput. Sci. Appl.*, Vol.7, 2010, pp. 1-22.

[11] Shutan Hsieh, Ching-Long Su, Jeffrey Liaw., Fuzzy ART for the Document Clustering By Using Evolutionary Computation, *WSEAS Transactions on Computers*, Vol.9, Issue 9, September 2010,pp.1032-1041.

[12] Kim, H.,Lee, S.,Lee, B., Kang, S., Building Concept Network-Based User Profile for Personalized Web Search. ACIS-ICIS, *IEEE Computer Society* , 2010, pp.567-572

[13] Leung, K. W. T., Ng, W., Lee, D.L., A framework for personalizing web search with concept-based user profiles, *ACM Transactions on Internet Technology (TOIT)* , Vol.11, No.4, 2012

[14] Prabaharan, S., Wahidabanu, R.S.D, Ontological Approach for Effective Generation of Concept Based User Profiles to Personalize Search Results *", J. Comput. Sci*., Vol.8, 2012, pp. 205-215.

[15] Beeferman, D., Berger., A., Agglomerative clustering of a search engine query log, *Proc.of KDD '00* , 2000, pp.407–416.

[16] Indumathi, D., Chitra A.,Girthana, K. 2013. Search Query expansion using Genetic Algorithm based clustering, *Smart computing Review- Korea Academia Industrial Cooperation Society*, Vol.3, No.1, 2013, pp. 14-23.

[17] Yahoo! Webscope dataset: http://webscope.sandbox.yahoo.com/

[18] Jarvelin.K and Kekalainen, J, "Cumulative Gain based Evaluation of IR Techniques", *ACM Transactions on Information Systems*, vol. 20, No.4, 2002, pp. 422-446.

[19] Qiu, F., Cho, J., Automatic identification of user interest for personalized search. *International conference on World Wide Web*, ACM press, 2006.

[20] Dou, Z., Song, R., Wen J., Large scale Evaluation and Analysis of Personalized Search Strategies. *Proceedings of WWW 2007*, ACM press, 2007.

[21] Mostefai Abdelkader, Malki Mimoun, Boudchiha Djeloul. Locating Services in Legacy Software:Information Retrieval Techniques, Ontology and FCA Based Approach, *WSEAS Transactions on Computers*, Vol.11, Issue 1, January 2012, pp.20-26

[22] Bouidghaghen, O., Paul Sabatier , Tamine, L. , Boughanem, M., 2011. Context-Aware User's Interests for Personalizing Mobile Search, *12th IEEE International Conference on Mobile Data Management (MDM)*, 2011.

[23] Chang Liu, Nicholas, J., Implicit acquisition of context for personalization of information retrieval systems. CaRR '11, *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, ACM New York, NY, USA, 2011

[24] Chang Zhanfang , Ban Xiaojuan ,Yao Yuan , Chang Binghu ,Wu Di, Personalized information retrieval based on user interest state. *IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing.*, 2012

[25] Indumathi, D., Chitra A., A Collaborative Search with Query Expansion and Result Re-ranking. *IEEE Proc. of World Congress on Information and Communication Technologies*, Mumbai, doi:10.1109/WICT.2011.6141382., 2011

[26] Jie Yu , Fangfang Liu, A Short-term User Interest Model for personalized recommendation. *The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, 2010.

[27] Shen, X.,Tan, B., Zhai, C., Context-sensitive information retrieval using implicit feedback. *Proc. of SIGIR '05*. New York, NY, USA: ACM Press, 2005.