CHAPTER 3
PERSONALIZED WEB SEARCH WITH LOCATION AND TIME PREFERENCES

3.1 Introduction

Recent growth of internet users’ and technology causes a fabulous growth in the amount of information on the web. Information retrieval systems are critical for overcoming this problem of information overload and providing the information of interest to users of the systems. Through a short query by a user given in any search engine consisting of a few keywords describing their information need. Information Retrieval based on ‘word to word’ match of the query words with all documents in their document collection and return documents containing the words entered. Recovery is much harder due to the large and dynamic content on the web. Most important web search engines like Google usually provide 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 and synonymy (different words can be used to convey similar information like OOP and Object Oriented Programming) due to unsure nature of natural language. Different backgrounds of users with different interests of users and ambiguities in natural language, is likely that query words of two different users may appear exactly same even though information needs are vary. Retrieval systems carry out a ‘word to word’ match of the query words and it work in a “one size fits all” fashion using the same search procedure for all the users. This builds the current retrieval systems faraway from optimal. This inherent non-optimality is seen clearly in the following three cases:

(1) **When a query contains uncertain terms**: The 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 Information Retrieval 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 and there might not be sufficient information required to infer information necessitate of user. Consider a query like “SBH” can mean “State Bank of Hyderabad” or “Syracuse Behavioural Healthcare” among others. Information Retrieval systems return blend of results containing the exact word which might contain different developments. Knowledge of interests and/or location of the user could be supportive in collecting more information essential to understand the query.

(3) **Information need of the user changes**: A users information needs may change over time to time. A user may use keyword “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. User context information about user and query is necessary for improving the retrieval performance. To capturing and exploiting related user context information of a query to improve search accuracy. The fast hustle and large amounts of data available in these online settings have recently made it imperative to use automated data mining or knowledge discovery techniques to discover Web user profiles. These different modes of usage or the so-called mass user profiles can be discovered using Web usage mining techniques that can automatically take out frequent access patterns from the history of previous user click streams stored in Web log files. Considerable advances in Web usage mining, there have been no detailed studies presenting a fully integrated approach to mine a real Web site with the challenging characteristics of today’s Web sites, such as evolving profiles, dynamic content, and the availability of taxonomy or databases in addition to Web logs.
User profile is important role in web personalization process. The search results are obtained from search engines to the users’ preferences through personalized re-ranking of the search result in the personalization process. Personalization techniques have been proposed to model users’ content preferences via study of users’ clicking and browsing behaviors [49], [52], [54], [57]. Also, techniques have been proposed for personalizing web search based on location [56]. This research recognizes the importance of time based search in addition to location, content information and proposes to incorporate these preferences in user profiles. Data and the information available on the web are evolving huge in amount. Some of the data are useful while some are not. Search engine results too many pages that might not be of the user’s interest. User spends lots of time in searching data and information on the rich web. Usage of online material is playing an important role for the learners in self discovery learning process but it is difficult to find relevant pages from the web because of gigantic web pages. For example, an agent based intelligent personalized e-learning model. The objective of the model is to produce preferred, relevant and recommended personalized search which provides the result of the learner’s query based on the learners' interests and preferences by re-ranking search engine result pages. Search engine is developed with implicit feedback received from learner’s browser behavior pattern which is stored as web log history and also from
explicit feedback through opinion of the learner. Query provided by a learner as an input will be practiced in step by step manner by multiple agents to generate personalized re-ranked search engine pages. The agents like Query Expansion agent, First Tier Search Agent, Middleware Agent, Web log pre-processor Agent, Web Knowledge Discovery Agent and Opinion Agent and client server architecture are some considered models. The huge mass of users is unwilling to provide any explicit feedback on search results and their interest. So, a personalized search engine proposed for a large audience has to learn the user’s preference automatically without any explicit input by the users. This research focus towards the problem of how to learn a user’s search interest automatically based on their past click history and how to use the learned interest to personalize search results for future queries. There exist a number of important technical questions to be addressed to realize this goal. First, it want to develop a reasonable user model that captures how a user’s click history is related to his/her interest; a user’s interest can be learned through her click history only if they are correlated. Second, based on this model it requires to design an effective learning method that identifies the user’s interest by analyzing the user’s click history. Finally, need to develop an effective ranking mechanism that considers the learned interest of the user in generating the search result.

3.1.1 Web Pre-fetching and Web Caching

The usage of Internet over the years has extremely increased and the users are leveraging its benefits to access variety of services provided over this network. The enormous growth of Internet have to force, network load and access time have increased dramatically causing substantial delay in providing services to the user. To overcome these limitations, web mining is considered as an effective solution where it focuses on minimizing the response time when web documents are retrieved. Web pre-fetching loads web documents into a web cache, which the user is likely to access in the near future. It
requires predicting the list of web documents to be pre-fetched for satisfying the user requests.

### 3.1.2 Web Log Mining

Web log mining is dealt with web log file analysis began with the reason to offer to Web site administrators a way to ensure adequate bandwidth and server capacity to their organization. The analysis made great advances with the passing of time, and now e-companies seek ways to use Web log files to obtain information about visitor profiles and buyers activities. Personalization is a possibility for the success of the evolving of a Web infrastructure. Web sites are created and adapted to make contents more easily accessible, using profiles found to make recommendations or to target users with ad hoc advertising. Web log files tell about usage history and information about a user, permitting to know his tastes and information needs from a web site.

### 3.2 METHODOLOGY

The methodologies used in this research for personalized web search is presented in this section. The methodologies used here is SpyNB method along with RSVM for re-ranking the search results according to the user preferences which be performed better than the usage of Joachims method. This section dealt with preference mining algorithms are namely Joachims’ method and SpyNB method are used to adopt in personalization framework.

#### 3.2.1 Joachim’s Method

In Joachim’s method [49] supposes that a user would scan the search result list from top to bottom. If a user skips a document $d_j$ at rank $j$ but clicks on document $d_i$ at rank $i$. 

50
Where \( j < i \), he/she must have read \( d_j \)'s web snippet and decided to skip it. Thus, Joachim’s method concludes that the user prefers \( d_i \) to document \( d_j \) (denoted as \( d_j < r' d_i \), where \( r' \) is the user’s preference order of the documents in the search result list).

<table>
<thead>
<tr>
<th>Links</th>
<th>The list of search results with titles, abstracts and URLs of Web pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Clicked)</td>
<td></td>
</tr>
<tr>
<td>( d_2 )</td>
<td>Apple - QuickTime - Download Visit the Apple Store online or at retail locations <a href="http://www.apple.com/quicktime/download/">http://www.apple.com/quicktime/download/</a></td>
</tr>
<tr>
<td>( d_3 )</td>
<td>Apple - Fruit Apples have a rounded shape with a depression at the top . . . <a href="http://www.hort.purdue.edu/ext/senior/fruits/apple1.htm">http://www.hort.purdue.edu/ext/senior/fruits/apple1.htm</a></td>
</tr>
<tr>
<td>( d_4 )</td>
<td>Apple. Mac Welcome . . . member specials throughout the year. See... <a href="http://www.mac.com/">http://www.mac.com/</a></td>
</tr>
<tr>
<td>(Clicked)</td>
<td></td>
</tr>
<tr>
<td>( d_5 )</td>
<td>A brief history of the company that changed the computing world . . . <a href="http://www.apple-history.com/">http://www.apple-history.com/</a></td>
</tr>
<tr>
<td>( d_7 )</td>
<td>Adams County Nursery, apple trees One of the most widely planted apple cultivars worldwide. <a href="http://www.acnursery.com/apples.htm">http://www.acnursery.com/apples.htm</a></td>
</tr>
<tr>
<td>( d_8 )</td>
<td>Apple–Support for most Apple products provided by Apple Computer <a href="http://www.info.apple.com/">http://www.info.apple.com/</a></td>
</tr>
<tr>
<td>(Clicked)</td>
<td></td>
</tr>
<tr>
<td>( d_{10} )</td>
<td>ROSE APPLE Fruit Facts - The rose apple is too large to make a suitable container plant. . . . <a href="http://www.cfgrg.org/pubs/fi/roseapple.html">http://www.cfgrg.org/pubs/fi/roseapple.html</a></td>
</tr>
</tbody>
</table>

Table 3.1 Joachim’s click through examples
Consider the click through examples listed in above table 3.1 to represent Joachims’ algorithm. All the preferences identified by Joachims’ algorithm are shown in Table 3.2 as follows.

<table>
<thead>
<tr>
<th>Preferences containing $d_1$</th>
<th>Preferences containing $d_4$</th>
<th>Preferences containing $d_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty set</td>
<td>$d_2 &lt; q d_4$</td>
<td>$d_2 &lt; q d_8$</td>
</tr>
<tr>
<td></td>
<td>$d_3 &lt; q d_4$</td>
<td>$d_3 &lt; q d_8$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_4 &lt; q d_8$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_5 &lt; q d_8$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_6 &lt; q d_8$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_7 &lt; q d_8$</td>
</tr>
</tbody>
</table>

**Table 3.2 Pair wise preferences identified by Joachims’ from the click through data**

Joachim’s algorithm explains the problem of penalizing high ranking links [Deng et al. 2004], which means that the high-ranking links (e.g., $d_1$, $d_2$) are more likely to be ‘less preferred’ compared to the low-ranking links (e.g., $d_9$, $d_{10}$). Consider the preference example shown in Table 3.2 Links $d_1$ and $d_4$ are both clicked links; however $d_1$ appears on the right -hand side of the preferences (meaning they are ‘preferred’ by the user) less often than $d_8$ does ($d_1$, 0 times; $d_8$, five times). On the other hand, links $d_2$ and $d_4$ are both unclicked links; however, $d_2$ appears on the left - hand side of the preferences (meaning “not preferred” by the user) more often than $d_9$ does ($d_2$, twice; $d_9$, 0 times). This explains the problem of over penalizing the high-ranking links.
3.2.2 Inadequacy of Existing Algorithms

Although Joachim’s algorithm is easy and competent their removal of preference pairs resulting from the stern scan order statement may not be entirely correct. The users’ behavior may be vary depends on the method of approach differs. For example, Joachim assumes that the user scans the search results strictly from top to bottom. There is possibility for skip several results without examining them carefully and clicks on a link at a lower rank. However, Joachim would conclude that these skipped links are considered as uninteresting to the user but in fact that could only say that whether these links are interesting to the user or not is unknown. As a result, the preferences identified by Joachim may not reflect users’ preferences accurately.

3.2.3 SpyNB Method

Similar to Joachim’s method, SpyNB [44] learns user behavior models from preferences extracted from click through data. SpyNB assumes that users would only click on documents that are of interest to them. Thus, it is reasonable to treat the clicked documents as positive samples. However, unclicked documents are treated as unlabeled samples because they could be either applicable or irrelevant to the user. Based on this understanding of clickthroughs, the problem becomes how to predict from the unlabeled set reliable negative documents which are irrelevant to the user. To do this, the spy technique incorporates a novel voting procedure into Naive Bayes classifier [51].
The SPYNB Algorithm

Input:

- $P$ - a set of positive examples;
- $U$ - a set of unlabeled examples;
- $T_v$ - a voting threshold;

Output:

- $PN$ - the set of predicted negative examples

Procedure:

1. $PN_1 = PN_2 = \cdots = PN_{P} = \{\}$ and $PN = \{\}$;
2. for each example $p_i \in P$ do
3. \hspace{1em} $P_s = P - \{p_i\}$;
4. \hspace{1em} $U_s = U \cup \{p_i\}$;
5. \hspace{1em} Assign each example in $P_s$ the class label 1;
6. \hspace{1em} Assign each example in $U_s$ the class label -1;
7. \hspace{1em} Train a Naïve Bayes on $P_s$ and $U_s$ using Algorithm 1;
8. \hspace{1em} Predict each example in $U_s$ using trained Naïve Bayes;
9. \hspace{1em} Spy threshold $T_s = Pr(+|p_i)$;
10. for each $u_j \in U$ do
11. \hspace{1em} if $Pr(+|u_j) < T_s$ then
12. \hspace{2em} $PN_i = PN_i \cup \{u_j\}$;
13. \hspace{1em} end if
14. end for
15. end for
16. for each $u_j \in U$ do
17. \hspace{1em} $Votes = \text{the number of } PN_i \text{ such that } u_j \in PN_i$
18. \hspace{1em} if $Votes > T_v \cdot |P|$ then
19. \hspace{2em} $PN = PN \cup \{u_j\}$;
20. \hspace{1em} end if
21. end for

The result discussed in details about the SpyNB method in [52]. Consider Let $P$ be the positive set, $U$ the unlabeled set and $PN$ the predicted negative set ($PN \subseteq U$) obtained from the SpyNB method. SpyNB assumes that the user would always prefer the positive set rather than the predicted negative set. User preference pairs can be discussed as follows.

$$d_i < d_j, \forall i \in P, j \in PN$$
3.3 Web search with Content, Location and Time

3.3.1 Concept Extraction

Personalization approach used in this research is based on concepts of profile the interests and preferences of a user. Therefore, an issue have to address is how to ‘extract’ and ‘represent’ concepts from search results of the user. Ontology-based, multi-facet (OMF) profiling method is considered for the proposed research, in which concepts can be further classified into different types, such as content concepts, location concepts, name entities, dates etc. As an important first step, this research focus on three major types of concepts, namely, content concepts, location concepts and time concepts. In a content concept, a keyword or key-phrase in a Web page is defines the content of the page, whereas a location concept refers to a physical location related to the page. And a time concepts reflects how a user interest or preference changes over a certain period of time.

3.3.2 Content Ontology

A keyword/phrase be presents frequently in the web-snippets arising from the query q, it represents an important concept related to the query, as it co-exists in close proximity with the query in the top documents. It extracts all the keywords and phrases from the web-snippets arising from q. After obtaining a set of keywords/phrases \( (c_i) \), the following support formula is used to measure the interestingness of a particular keyword/phrase \( c_i \) with respect to the query q, which is motivated by the well-known problem of finding frequent item sets in data mining [47].

\[
\text{Support} (c_i) = \frac{sf(c_i)}{n} \cdot |c_i|
\]

Where \( sf(c_i) \) is the snippet frequency of the keyword / phrase \( c_i \) (i.e.) the number of web-snippets containing \( c_i \), n is the number of web-snippets returned and \( |c_i| \) is the number of terms in the keyword/phrase \( c_i \). If the support of a keyword/phrase \( c_i \) is higher
than the thresholds \( s = 0.034 \) in this experiments, treat \( c_i \) as a concept for the query \( q \).

This research focuses the use of ontologies to maintain concepts and \( c_i \) their relationships extracted from search results. That captures the following two types of relationships for content concepts:

**Similarity**: Two concepts which coexist a lot on the search results might represent the same topical interest. If coexist \( (c_i, c_j) > \delta_1 \) \( (\delta_1 \) is a threshold), then \( c_i, c_j \) are considered as similar.

**Parent-Child Relationship**: More specific concepts often appear with general terms, while the reverse is not true. If \( \Pr(c_j | c_i) > \delta_2 \) \( (\delta_2 \) is a threshold), and mark \( c_i \) as \( c_j \)'s child.

### 3.3.3 Location Ontology

The approach mentioned in this research for extracting location concepts is entirely different from content concepts. A document usually embodies only a few location concepts first. And as a result, very few of them co-occur with the query terms in selected web snippets. To overcome this problem the extracts using location concepts from the full documents is considered first. Second, due to the small number of location concepts embodied in documents, the similarity and parent-child relationship cannot be accurately derived statistically. The physical relationships among locations have been captured as facts of search methods. Thus, it creates a predefined location ontology consisting of about 17,000 city, province, region, and country names obtained from [34] and [36]. In the location ontology, that organizes all the cities as children under their provinces, all the provinces as children under their regions, and all the regions as children under their countries. The statistics of location ontology are provided in Table 3.2. The location ontology extraction method first extracts all of the keywords and key-phrases from the documents returned for \( q \). If a keyword or key-phrase in a retrieved document \( d \) matches
a location name in predefined location ontology, it will be treated as a location concept of d.

<table>
<thead>
<tr>
<th>No. of Countries</th>
<th>6</th>
<th>Total No of Nodes</th>
<th>16890</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Regions</td>
<td>180</td>
<td>Country-region Edges</td>
<td>195</td>
</tr>
<tr>
<td>No. of States</td>
<td>6700</td>
<td>Region-State Edges</td>
<td>1960</td>
</tr>
<tr>
<td>No. of Cities</td>
<td>10005</td>
<td>Province-City Edges</td>
<td>14895</td>
</tr>
</tbody>
</table>

**Table 3.3 Statistics of the Location ontology**

3.3.4 Content and Location Entropies

The amount of queries differs from its associated amount of different content and different location information. Consider the example; the queries such as overseas study may have strong associations to a large number of location concepts. Queries such as programming tend to be content-oriented with only weak association to location concepts (i.e., most concepts, such as books and software tools, related to computer programming are location independent). Some queries can be well-off in both content and location information. (e.g. Shopping). The content and location properties of a query are formally characterized by the use of entropy to estimate the amount of content and location information retrieved by a query. Entropy indicates the uncertainty associated with the information content of a message from the receiver’s point of view as stated in information theory [45]. Entropy can be employed in a similar manner to denote the uncertainty associated with the information content of the search results from the user’s point of view in search engines. Content and location information that define two entropies, namely, content entropy and location entropy to measure the uncertainty associated with the content and location information of the search results. The information entropy of a discrete random variable X is defined as:
Where \( n \) are the possible values \( x_1, x_2, \ldots, x_n \) of \( X \) and \( p(x_i) = \Pr(x=x_i) \), this research adopts the above formula to compute the content and location entropies of a query \( q \) (i.e. \( H_C(q) \) and \( H_L(q) \)) as follows:

\[
H_C(q) = - \sum_{i=1}^{k} p(c_i) \log p(c_i) \quad H_L(q) = - \sum_{i=1}^{m} p(l_i) \log p(l_i)
\]

Where \( k \) is the number of content concepts \( C = c_1, c_2, \ldots, c_k \) extracted, \( |c_i| \) is the number of search results containing the content concepts \( c_i \), \( |C| = |c_1| + |c_2| + \ldots + |c_k| \), \( p(c_i) = \frac{|c_i|}{|C|} \), \( m \) is the number of location concepts \( L = l_1, l_2, \ldots, l_m \) extracted, \( |l_i| \) is the number of search results containing location concepts \( l_i \), \( |L| = |l_1| + |l_2| + \ldots + |l_k| \), and \( p(l_i) = \frac{|l_i|}{|L|} \).

### 3.3.5 Click Content and Location Entropies

Content and location entropies introduce ‘click content entropy’ and ‘click location entropy’ to point out the diversity of a user’s interest on the content and location information returned from a query. The entropy equations for click content and location concepts are similar to Equation (3), but only the clicked pages, and hence the clicked concepts, are considered in the formula. Since the click entropies reflect the user’s actions in response to the search results, that can be used as a signal of the diversity of the user’s interests. Formally, the click content entropy \( H_C = (q,u) \) and click location entropy \( H_L = (q,u) \) of a query \( q \) submitted by the user \( u \) are defined as follows:

\[
H_C(q,u) = - \sum_{i=1}^{t} p(\tilde{c}_i u) \log p(\tilde{c}_i u) \\
H_L(q,u) = - \sum_{i=1}^{t} p(\tilde{l}_i u) \log p(\tilde{l}_i u)
\]
where \( t \) is the number of content concepts clicked by the user \( u \), \( \overline{C}_u = \overline{c}_1^u, \overline{c}_2^u, \ldots, \overline{c}_t^u \), \( |\overline{c}_{i,u}| \) is the number of times that content concept \( c_i \) has been clicked by the user \( u \),

\[
\overline{C}_u = \overline{c}_1^u + \overline{c}_2^u + \ldots + \overline{c}_t^u, \quad p(c_i,u) = \frac{|\overline{c}_{i,u}|}{|\overline{C}_u|}, \quad v \text{ is the number of location concepts}
\]

\[
\overline{L}_u = \overline{l}_1^u, \overline{l}_2^u, \ldots, \overline{l}_v^u \text{ clicked by u, } |\overline{l}_{i,u}| \text{ is the number of times that the location concept } l_i \text{ is being clicked by the user u, } \overline{L}_u = |\overline{l}_1^u| + |\overline{l}_2^u| + \ldots + |\overline{l}_v^u| \text{ and } p(l_i,u) = \frac{|\overline{l}_{i,u}|}{|\overline{L}_u|}.
\]

### 3.3.6 Incorporating time based personalization

A personalization system that handles user’s profiles, content and location entropies and application of the user’s profiles on that content and location entropies is the first step towards incorporating time in the personalization process. A system can be able to work as follows

1. User’s preference or interest is captured according to user time zones and it maintains the click through ontology along with time zone.
2. User’s device profile is collected and maintained. It could be implemented as part of the “user profile management” component.
3. Describe the available content and location entropies. Having a “content and location ontology”
4. Apply the user’s preferences on the content and location ontology in order to select the desired content and location features.
5. Training process takes place followed by update the user profile according to the user preferences

Time based personalization technique that needs to know how the user’s preferences vary over the 24 hour per day cycle. Time zone represents various time that suggests dividing
the day into different time-zones. This study can be takes place on daily routine of users and then split it into number of time zones based on the user’s activities for every day.

Figure 3 b: Time zones with different time limit

The list of time zones prepared for different time limits is shown in figure 3b for a particular person’s activities on working day.

Figure 3 c: Daily activities of a person split into different time-zones

Figure 3c shows an example of the daily cycle of a user and how it could be split into different time-zones.
Table 3.4 Time zones for corresponding content and location click through data

By giving preferences to time zones the user profile must be improved in order to contain all the (newly) required/available metadata. The user profile that holds his preferences must be easily extendible, easy to handle (retrieve or store preferences) and somewhat standardized by using time zones.

By dividing the day into different time-zones, this research drastically reduces the possible combinations between time and user’s preferences, keeping the design scalable. Having the time-zones on one hand, and the preferences weights on the other, enable to capture the required information. It just needs to record how the weight of each preference changes over each time-zone.
And also needs to assign a weight to each of the users’ preferences to differentiate among them. Weights in effect, show how much a user likes or dislikes to a specific content and location information. Figure 3e shows how the users’ interest for a given concept (e.g. restaurants) time changes during a day with different weight.
A weighted system with time zones allows the user to declare what and when is more important to him. It can use the weight association to identify each user’s experience. For a new experience of a user by just adding additional set of weights it can record the user’s new preferences. For example if the user goes on vacation then his preference would be on vacation is mandatory. Changes is normal from his daily experience is the composition of his day cycle. During the vacation the user’s profile would have increased weights for content related to pastime and decreased weights for job related content. As a result, put together the user experience is like building a users’ profile for his every activities cycle. Thus, the obtained knowledge about the user that need not to store in a separate profile and it replicate the weight structure of the normal daily cycle with different values resulting dynamic environment of a user and his continuous activities.

The user profile and the content description are based on the same ontology it achieves improved effectiveness during the content selection. It compares each content information node (content service) description with the users’ profile that results the comparison is made on the characteristics between matching categories of the content description and the users’ profile for the current time zone. The preference weight is attained for each attribute from the user profile beside with the weight for the given category. For more than one user’s experiences also follows the same method. The weight set associated with the current users’ experience is loaded in user profile. A minute by minute updates is recorded in a dynamic user profile. The timing and users’ experience affects the personalization process by permitting the execution of active user profiles.
The process of averaging the retrieved weights is the final step in the content selection and assigning a selection weight to each content node. The results are listed in user sorted order based on weight. Nodes with lower weight than a user selected threshold are entirely not considered. The enhanced user profile is created with such time–zone preferences and user preferences. The user profile which holds his preferences can be simply extendible, easy to handle and standardized.

3.4 Personalized Ranking Functions

In this research Ranking SVM [49] is working in personalization approach to study the users’ preferences. A set of content concepts and a set of location concepts are extracted from the search result for a considered query in a selected document. Each document represented by a feature vector and it can be treated as a point in the attribute space. Using click through data as the input, RSVM aims at finding a linear ranking function, which holds for as many document preference pairs as possible. In this research, an adaptive implementation, SVM light available, is used for the training purpose. It outputs
a ‘content weight vector’ $W_{C,q,u}$ and a ‘location weight vector’ $W_{L,q,u}$, which best describes the user interests based on the users’ content and location preferences extracted from the user click through, respectively. The two issues in the RSVM training process: 1) how to extract the feature vectors for a document; 2) how to combine the content and location weight vectors into one integrated weight vector are discussed as follows

### 3.4.1 Extracting Features for Training

Content feature vector denoted by $\phi_c = (q,d)$ and location feature vector denoted by noted by $\phi_L = (q,d)$ are defined two feature vectors to represent documents. The feature vectors are extracted by taking into account the concepts existing in a document and other related concepts in the ontology of the query. The extraction of content feature vector and location feature vector are defined formally as follows

**Content Feature Vector**

If a content concept $c_i$ is in a web-snippet $s_k$, their values are incremented in the content feature vector $\phi_c = (q,d)$ with the following equation:

$$\forall c_i \in s_k; \phi_C(q,d_k)[c_i] = \phi_C(q,d_k)[c_i] + 1$$

The content concepts $c_i$ that are related to the content concept $c_j$ (either they are similar or $c_j$ are the ancestor/descendant/sibling of $c_i$) in the content ontology, they are incremented in the content feature vector $\phi_c = (q,d_k)$ based on the following equation:

$$\forall c_i \in s_k; \phi_C(q,d_k)[c_j] = \phi_C(q,d_k)[c_j] + sim_{R}(c_i, c_j) + \text{ancestor}(c_i,c_j) + \text{descendant}(c_i,c_j) + \text{sibling}(c_i,c_j)$$
Location Feature Vector

A location concept \( l_i \) is in a web-snippet \( d_k \), their values are increased in the location feature vector \( \phi_L = (q, d_k) \) with the following equation:

\[
\forall l_i \in d_k, \phi_L(q, d_k)[l_i] = \phi_L(q, d_k)[l_i] + 1
\]

For other location concepts \( l_i \) that are related to the concept \( l_i \) (\( l_i \) is the ancestor/descendant/sibling of \( l_j \)) in the location ontology, they are incremented in the location feature vector \( \phi_L = (q, d_k) \) according to the following equation.

\[
\forall l_i \in d_i: \quad \phi_L(q, d_k)[l_j] = \phi_L(q, d_k)[l_j] + \text{ancestor}(l_i, l_j) + \text{descendant}(l_i, l_j) + \text{sibling}(l_i, l_j)
\]

3.4.2 Combining Weight Vectors

The content feature vector \( \phi_C = (q, d) \) within the document preferences get from SpyNB approaches are serve up as input to RSVM training to get the ‘content weight vector’ \( w_{C,q,u} \). ‘The location weight vector’ \( w_{L,q,u} \) is attained similarly using the location feature vector \( \phi_L = (q, d) \) and the document preferences. The two weight vectors are \( w_{C,q,u} \) and \( w_{L,q,u} \) stands for the content and location user profiles for a user \( u \) on a query \( q \) in ontology based, multi-facet (OMF) by user profiling methods.

This research uses the formula to combine the two weight vectors, \( w_{C,q,u} \) and \( w_{L,q,u} \) to optimize the personalization effect. According to the values of the personalization effectiveness parameters \( e_C(q, u) \) and \( e_L(q) \), the final weight vector
\( w_{q,u} \) for user ranking is obtained. The two weight vectors, \( \overrightarrow{w_{C,q,u}} \) and \( \overrightarrow{w_{L,q,u}} \)

are first normalized by the combination.

\[
\overrightarrow{w_{q,u}} = \frac{\varepsilon_C(q, u) \cdot \overrightarrow{w_{C,q,u}}}{\varepsilon_C(q, u) + \varepsilon_L(q, u)} + \frac{\varepsilon_L(q, u) \cdot \overrightarrow{w_{L,q,u}}}{\varepsilon_C(q, u) + \varepsilon_L(q, u)}
\]

Let \( e(q, u) = \frac{e_C(q, u)}{e_C(q, u) + e_L(q, u)} \), and the following formula is obtained.

\[
\overrightarrow{w_{q,u}} = e(q, u) \cdot \overrightarrow{w_{C,q,u}} + (1 - e(q, u)) \cdot \overrightarrow{w_{L,q,u}}
\]

The final weight vector \( \overrightarrow{w_{q,u}} \) is computed and a linear ranking function is adopted for rank alteration of future search results. The documents in the future search will be ranked as the given formula.

\[
f(q, d) = \overrightarrow{w_{q,u}} \cdot \phi(q, d)
\]

Where \( q \) is a query, \( d \) is a document in the search results, \( \overrightarrow{w_{q,u}} \) is the weight vector, and \( \phi(q, d) \) is a feature vector representing the match between query \( q \) and document \( d \).

After performing RSVM for profile updation based on content, location according to time zones the effectiveness rate of personalization of web search can be improved.

### 3.5 RESULTS AND DISCUSSION

The experimental results of the proposed personalized web search based on content, location and time preferences are provided in this section. The experimentation is performed using the implemented prototype.

#### 3.5.1 Input Data

20 users are invited to submit totally 100 test queries through search engine. Totally 100 test queries consists 10 categories. The 20 users are assigned in 5 test queries are
randomly picked from the 10 dissimilar categories. The users are prearranged the tasks to find results that are pertinent to their interests. The clicked results are stored in the click through record along with the time zones and are pleasured as optimistic samples in RSVM training.

3.5.2 Evaluation of Performance rate

The Performance rate of the proposed method is compared with the existing methods like Joachim’s, SpyNB, SVM and the chart depicts the variation among the results of clicked results in terms of accuracy. 5 test queries are casually selected from the 10 dissimilar categories of 20 unlike users. The users are given the tasks to find results that are relevant to their interests.

<table>
<thead>
<tr>
<th>Number of Training Queries</th>
<th>No. of Results</th>
<th>Irrelevant result</th>
<th>Accuracy by Joachim’s method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>76</td>
<td>9</td>
<td>92</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>14</td>
<td>82</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>6</td>
<td>87</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>7</td>
<td>86</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>10</td>
<td>83</td>
</tr>
</tbody>
</table>

Table 3.5: List of relevant links in top 20 links by Joachim’s method

The above table 3.5 shows list of the performance accuracy for Joachim’s method.

<table>
<thead>
<tr>
<th>Number of Training Queries</th>
<th>No. of Results</th>
<th>Irrelevant result</th>
<th>Accuracy by SpyNB method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75</td>
<td>7</td>
<td>91</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>12</td>
<td>79</td>
</tr>
<tr>
<td>3</td>
<td>44</td>
<td>8</td>
<td>82</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>8</td>
<td>84</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>12</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 3.6: List of relevant links in top 20 links by SpyNB method
The table 3.6 shows the performance accuracy for SpyNB method. The following graphs show the result of accuracy of relevant link through Joachim’s and SpyNB methods respectively. To produce the graph, the x-axis shows the number of training queries and the y-axis shows the percentage of accuracy.

![Accuracy by Joachim’s Method](image1)

**Figure 3 g: User Query Vs Accuracy by Joachim’s Method**

![Accuracy by SpyNB Method](image2)

**Figure 3 h: User Query Vs Accuracy by SpyNB Method**
<table>
<thead>
<tr>
<th>No of Training Queries</th>
<th>No. of Results</th>
<th>Irrelevant result</th>
<th>Accuracy by SVM method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>77</td>
<td>6</td>
<td>92</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>8</td>
<td>86</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>5</td>
<td>89</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>7</td>
<td>86</td>
</tr>
<tr>
<td>5</td>
<td>58</td>
<td>9</td>
<td>84</td>
</tr>
</tbody>
</table>

**Table 3.7: Statistics to display the relevant links in top 20 links by SVM Method**

The table 3.7 shows the performance accuracy by SVM method for the relevant top 20 links from selected 5 queries.

<table>
<thead>
<tr>
<th>Number of Training</th>
<th>No. of Results</th>
<th>Irrelevant Result</th>
<th>Accuracy by RSVM method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75</td>
<td>3</td>
<td>96</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>4</td>
<td>93</td>
</tr>
<tr>
<td>3</td>
<td>44</td>
<td>4</td>
<td>91</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>3</td>
<td>94</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>5</td>
<td>92</td>
</tr>
</tbody>
</table>

**Table 3.8: Statistics to display the relevant links in top 20 links by proposed time based RSVM Method**

The table 3.8 shows the performance accuracy by proposed time based RSVM method for the relevant top 20 links from 5 selected queries of 20 users.
Figure 3 i: Comparison between SVM and time based RSVM for accuracy

The above figure 3i shows that the result of accuracy of 5 selected user queries are compared with the SVM and proposed time based RSVM. To produce the chart, the full data set is split randomly into training and a test set. The x-axis shows the number of training queries. The y-axis shows the percentage of accuracy. The result depicts RSVM training set produces more accuracy than SVM.

Figure 3 j: Performance rate comparison of a proposed time based search for a User Group (1)
The above figures 3 j, 3 k and 3 l show that the result of accuracy in terms of performance rate of selected user groups (1), user groups (2) and user groups (3) (each set of 5 users) respectively. All user queries are compared with the Joachim’s, SpyNB, SVM and proposed RSVM. To produce the chart, the full data set is split randomly into...
training and a test set. The x-axis shows the number of training queries. The y-axis shows the percentage of accuracy. The result depicts RSVM training set produces more accuracy than other training sets even for different user groups.

<table>
<thead>
<tr>
<th>User (1 to 20)</th>
<th>Existing Method</th>
<th>Time based RSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>86</td>
<td>93</td>
</tr>
<tr>
<td>U2</td>
<td>82</td>
<td>91</td>
</tr>
<tr>
<td>U3</td>
<td>83</td>
<td>90</td>
</tr>
<tr>
<td>U4</td>
<td>79</td>
<td>89</td>
</tr>
<tr>
<td>U5</td>
<td>82</td>
<td>92</td>
</tr>
<tr>
<td>U6</td>
<td>80</td>
<td>89</td>
</tr>
<tr>
<td>U7</td>
<td>85</td>
<td>92</td>
</tr>
<tr>
<td>U8</td>
<td>75</td>
<td>89</td>
</tr>
<tr>
<td>U9</td>
<td>78</td>
<td>91</td>
</tr>
<tr>
<td>U10</td>
<td>79</td>
<td>93</td>
</tr>
<tr>
<td>U11</td>
<td>81</td>
<td>94</td>
</tr>
<tr>
<td>U12</td>
<td>82</td>
<td>92</td>
</tr>
<tr>
<td>U13</td>
<td>80</td>
<td>94</td>
</tr>
<tr>
<td>U14</td>
<td>84</td>
<td>93</td>
</tr>
<tr>
<td>U15</td>
<td>79</td>
<td>89</td>
</tr>
<tr>
<td>U16</td>
<td>75</td>
<td>88</td>
</tr>
<tr>
<td>U17</td>
<td>74</td>
<td>89</td>
</tr>
<tr>
<td>U18</td>
<td>73</td>
<td>83</td>
</tr>
<tr>
<td>U19</td>
<td>72</td>
<td>89</td>
</tr>
<tr>
<td>U20</td>
<td>75</td>
<td>90</td>
</tr>
</tbody>
</table>

**Table 3.9: Average Performance Rate for 20 different users**

The above table 3.9 lists the average performance rate of 20 different users. By using this predictive performance rate of RSVM is calculated and compared with existing methods.

**Figure 3 m: Predictive performance rate of a proposed time based Ranking SVM in terms of Users’ preferences**
The result shows in figure 3 that personalizing web search based on content, location and time preferences provide high performance rate in terms of getting relevant information based on users’ interest or preferences. Each point is an average over 10 (5-20 training queries) / 20 (40-80 training queries) different test/training splits. The graph shows that the Ranking SVM can learn regularities in the preferences. The performance rate provided by the proposed personalized web search is higher than the existing one during Google search. These results provide a first proof of concept and justify a larger-scale experiment with multiple users.

Summary

This research focuses about some problems in real world information access from the web through web search engines. The main problem with web search is as follows: There is too much of information available on the web. Query words used by users often are confusing, ambiguous and sometimes are poor descriptors of information need (‘SBH’ can mean “State Bank of Hyderabad” or “Syracuse Behavioral Healthcare” among others); users are often not patient enough to see long list of results given by search engines to find relevant information. It has been observed that users typically only view top few results, usually top 5 or 10, some times 20 and much fewer times 30 and so on. In this scenario, web search can be made more useful, effective and less burdensome to users if search results are customized to each individual user by considering individual users’ idiosyncrasy. This could possibly help the user to find out the relevant document easily from having to scroll through a long list of results.

This chapter focuses an ontology based web user personalization and search with content, location and time preferences. This approach extracting the concepts which are related to the users’ query are divided into content and location concepts which are useful to create two ontologies are such as content and location. Furthermore, the user first choices
according to the time zone are pull out which will make click through ontology. The ontology of users’ profile will be fashioned and efficient using RSVM. The investigational result shows that anticipated personalization move toward offers higher performance rate measure up to with conventional methods. In current scenario, personalized search has create a center of attention to interest in the research group of people as a means to reduce search uncertainty and return results that are likely to be interesting to a particular user and therefore given that more effective and efficient information access. This chapter deals with a proposed web user personalization and search technique with content, location and time preference of a user which helps user to get highly relevant information according to his/her interest.