Time based Web User Personalization and Search

D.Dhanalakshmi
M.Phil Scholar and Assistant Professor
Department of Computer Science
Dr.N.G.P Arts and Science College, Coimbatore-48
Tamil Nadu, India

R.Kousalya
Ph.D Scholar, Manonmaniam Sundaranar University, Tirunelveli
Head, Department of Computer Applications, Dr.N.G.P Arts and Science College, Coimbatore-48, Tamil Nadu, India

V.Saravanan
Professor and Director, Department of Computer Applications,
Sri Venkateswara College of Computer Applications and Management, Coimbatore-105, Tamil Nadu, India

ABSTRACT
The information on the World Wide Web is growing without bound. Users may have very diversified preferences in the pages they target through a search engine. It is therefore a challenging task to adapt a search engine to suit the needs of a particular user. In mobile search, the interaction between users and mobile devices are constrained by the small form factors of the mobile devices. To reduce the amount of user's interactions with the search interface, an important requirement for mobile search engine is to be able to understand the users' needs and preferences on that instant and deliver highly relevant information to the users. To effectively aid this task, we propose an efficient approach for web user personalization and search. In our approach, user's interests and preferences according to time are extracted by mining time of access, search results and their clickthroughs. User profile will be created and updated using RSVM training. Experimental result shows that, personalization according to time preference improve the effectiveness rate of personalization and search.

General Terms
Mobile users, Web user Personalization

Keywords
Web Mining, Ontology, Entropy, Time zones, RSVM

1. INTRODUCTION
There has been a tremendous growth in the amount of information on the web. Information retrieval systems are critical for overcoming this information overload and providing the information of interest to users of the systems. Users typically pose a short query consisting of a few keywords describing their information need. Information Retrieval systems [1] perform a 'word to word' match of the query words with all documents in their document collection and return documents containing the words entered. Retrieval in a web scenario is much harder due to the large and dynamic content on the web. Major web search engines usually cater to hundreds of millions of users and hundreds of millions queries every day. It is very unlikely that the millions of users are similar in interests and search for similar information. Also, it is probable that the query words entered by users exhibit polysemy (same word used in different senses like 'java' can be used to mean Java programming language or Java islands in Indonesia) and synonymy (different words can be used to convey similar information like OOP and Object Oriented Programming) due to ambiguous nature of natural language. Therefore, given different backgrounds of users, different interests of users and ambiguities in natural language, it is very likely that query words of two different users may appear exactly same even though information needs are different.

However, current retrieval systems perform a 'word to word' match of the query words and work in a "one size fits all" fashion using the same search procedure for all the users. This makes the current retrieval systems far from optimal. This inherent non-optimality is seen clearly in the following three cases: (1) when a query contains ambiguous terms: Different users may use exactly the same query (e.g, "Java") to search for different information (e.g., the Java island in Indonesia or the Java programming language), but existing IR systems return the same results for these users. Without considering the actual user, it is impossible to know which sense "Java" refers to in a query. (2) When a query contains partial information: A query can contain an acronym or a shorter usage of a longer phrase. Then there might not be sufficient information required to infer information need of user. For example a query like "SBH" can mean "State Bank of Hyderabad" or "Syracuse Behavioral Healthcare" among others. Existing IR systems return mixture of results containing the exact word which might contain different expansions. Knowledge of interests and/or location of the user could be helpful in gathering more information required to understand the query. (3) When information need of the user changes: A users information needs may change over time. The same user may use "Java" sometimes to mean the Java Island in Indonesia and some other times to mean the programming language. Without recognizing the search context, it would be again impossible to recognize the correct sense. Thus using user context information about user and query is necessary for improving the retrieval performance. Indeed, personalized search essentially boils down to capturing and exploiting related user context information of a query to improve search accuracy. Current retrieval systems (or search engines) return a long list of results obtained by 'word to word' match with query words. However, it has been observed that users typically view only top few (usually 10) documents out of the long list of results returned by search engines. This requires retrieval systems to show the most relevant documents to a user on the top to improve user satisfaction with the search engine. However, without knowledge about the user context, this task is difficult to do because "relevance" of a document depends on the individual user and the individual query. The goal of personalized search is to customize or personalize the search results returned by a search engine according to each individual user.

2. PROPOSED MODEL

Personalization of search results is very important to the future success of any search engine. Personalization is not however, a magical phenomenon or crystal ball interpretation. It is purely based on observed patterns, and resulting probabilities. Unlike desktop users, mobile users are a new and more demanding breed. Technology provided for the first group is often found lacking for the later. Personalization is such an example. To effectively aid this task, the solution of personalization and user profiling is often used. To accomplish web user personalization and search for the mobile users in our dissertation we consider two important factors such as content and location with time preference. For this task we consider [2] as a base work. The methodologies used here is SpyNB preference mining algorithm along with RSVM for re-ranking the search results according to the user preferences based on content, location and time. It offers better personalization because of considering the factors such as content and location with time preference.

2.1 User Personalization and Search

Our personalization approach is based on concepts related to time to profile the interests and preferences of a user. We focused on two major types of concepts, namely, content concepts, location concepts with time preference. A content concept, like a keyword or key-phrase in a Web page, defines the content of the page, whereas a location concept refers to a physical location related to the page [2]. A time preference reflects how a user interest or preference changes over a period of time.

2.2 An ontology for Content

We assume that if a keyword/phrase exists 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. Thus, our content concept extraction method first 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, which is inspired by the well-known problem of finding frequent item sets in data mining [3], is employed to measure the interestingness of a particular keyword/phrase c_i with respect to the query q:

$$\text{Support}(c_i) = \frac{sf(c_i)}{n} \cdot |c_i| \quad (1)$$

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 threshold s (s = 0.03 in our experiments), we treat c_i as a concept for the query q. As mentioned, we use ontologies to maintain concepts c_i and their relationships extracted from search results. We capture 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 $(c_i, c_j) > \delta_1$ (δ_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. Thus, if $\text{pr}(c_j | c_i) > \delta_2$ (δ_2 is a threshold), we mark c_i as c_j's child.

2.3 An Ontology for Location

Our approach for extracting location concepts is different from that for extracting content concepts. First, a document usually embodies only a few location concepts. As a result, very few of them co-occur with the query terms in web snippets. To alleviate this problem, we extract location concepts from the full documents. Second, due to the small number of location concepts embodied in documents, the similarity and parent-child relationship cannot be accurately derived statistically. Additionally, the geographical relationships among many locations have already been captured as facts. Thus, we create a predefined location ontology consisting of about 17,000 city, province, region, and country names obtained from [4] and [5]. In the location ontology, we organize 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 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 our predefined location ontology, it will be treated as a location concept of d.

2.4 Mining Content and Location Notion

Different queries may be associated with different amount of content and location information. For example, queries such as Overseas Study may have strong associations to a large number of location concepts. However, 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). Meanwhile, some queries (e.g. Shopping) can be rich in both content and location information. To formally characterize the content and location properties of a query, we use *entropy* to estimate the amount of content and location information retrieved by a query.

In information theory [6], *entropy* indicates the uncertainty associated with the information content of a message from the receiver's point of view. In the context of search engine, 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. Since we are concerned with content and location information only in this paper, we used two entropies, namely, content entropy and location entropy, to measure, respectively, 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:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \quad (2)$$

Where n is the possible values $\{x_1, x_2 \ldots x_n\}$ of X and $p(x_i)$ $=Pr(x=x_i)$ We adopt 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:

$$H_C(q) = -\sum_{i=1}^{k} p(c_i) \log p(c_i)$$

$$H_L(q) = -\sum_{i=1}^{m} p(l_i) \log p(l_i) \qquad (3)$$

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 $p(l_i) = \frac{|l_i|}{|L|}$.

2.5 Mining Clickthrough Data

As with content and location entropies, we introduce click content entropy and click location entropy to indicate, respectively, 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 Eq. (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, they can be used as an indication of the diversity of the user's interests. Formally, the click content entropy $H_{\overline{C}}=(q,u)$ and click location entropy $H_{\overline{L}}=(q,u)$ of a query q submitted by the user u are defined as follows:

$$H_{\overline{C}}(q,u) = -\sum_{i=1}^{t} p(\bar{c}_{iu}) \log p(\bar{c}_{iu})$$

$$H_{\overline{L}}(q,u) = -\sum_{i=1}^{v} p(\bar{l}_{iu}) \log p(c\bar{l}_{iu}) \qquad (4)$$

where t is the number of content concepts clicked by the user u, $\overline{C}_u = \{\overline{c_{1u}}, \overline{c_{2u}}, \ldots, \overline{c_{tu}}\}$, $|\overline{c_{iu}}|$ is the number of times that content concept c_i has been clicked by the user u, $|\overline{C}_u| = |\overline{c_{1u}}| + |\overline{c_{2u}}| + \ldots |\overline{c_{tu}}|$, $p(\overline{c_i}, u) = \frac{|\overline{c_{iu}}|}{|\overline{c_u}|}$, v is the number of location concepts $\overline{L}_u = \{\overline{l_{1u}}, \overline{l_{2u}}, \ldots, \overline{l_{vu}}\}$ clicked by u, $|\overline{l_{iu}}|$ is the number of times that the location concept l_i is being clicked by the user u, $|\overline{L}_u| = |\overline{l_{1u}}| + |\overline{l_{2u}}| + \ldots |\overline{l_{vu}}|$ and $p(\overline{l_i}, u) = \frac{|\overline{l_{iu}}|}{|\overline{l_u}|}$.

2.6 An Ontology for User Preference According to Time Zones

When we consider time while doing personalization it returns results based on times it perceives you are typically working. Having in place 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. In detail, such a system should be able to:

- Capture the user's preference or interest according to time zones and maintains the clickthrough ontology along with time zones.
- Capture the user's device profile. Again, this could be implemented as part of the "user profile management" component.
- Describe the available content and location entropies. Having a "content and location ontology"
- Combine the user's preferences for the particular time zone along with content and location ontology in order to select the desired content and location features for that time zone.
- Train and update the user profile according to the user preferences

To achieve time based personalization we need to know how the user's preferences change over the 24 hour day cycle. To represent time we suggest dividing the day into different time-zones. This is possible if we study the daily routine of our users and then split it into time zones based on the user's activities for each period.

Table 1 User's preferences for 24 hour per day cycle

Time Zone	User Preference
0-8 and 23-0	Rest
8-12	Work
12-14	Lunch
14-18	Work
18-21	Recreation
21-23	Dine out

By dividing the day in time-zones, we drastically reduce the possible combinations between time and user's preferences, keeping our design scalable.

From the user clickthrough data, clickthrough ontology for content and location will be created for all timezones.

The Proposed algorithm for time based personalization will be called as **Time based Personalization Algorithm** (TBPA) as follows:

Step 1: Define clickthrough ontology for each timezones during training

Step 2: If new query is submitted, the middleware (exits between user and search engine) extracts time from the system

Step 3: Extracted time is matched with each timezone, and the matching timezone's clickthrough ontology will be considered to identify user preference on that instance
Step 4: Search results will be re-ranked for the user
Step 5: If the user prefers the concept other than the top ranked concepts then equal weight (i.e. the weight of the existing top ranked documents) is assigned for the new concept
Step 6 : Next time when the user submits the query repeat the steps 3
Step 7: If the concepts having equal weights Middleware ask user which concept is preferred at this moment
Step 8: Based on the user response again weights will be updated for the concepts

2.7 Learning User Preference

1) Ranking SVM: Here is the algorithm for Ranking SVM

- Input Space:X
- Ranking Function f:X $\rightarrow$ R
- Ranking function

$$x_i \succ x_j \Leftrightarrow f(x_i;w) \rangle f(x_j;w)$$

- Linear Ranking function: $f(x;w)=\langle w,x\rangle$

$$\langle w,w^{(1)}-w^{(2)}\rangle > 0 \Leftrightarrow f(x^{(1)};w) > f(x^{(2)};w)$$

- Transforming to binary classification:

$$(\ddot{x}^{(1)}-\ddot{x}^{(2)},z), z=\begin{cases}+1 & x^{(1)} \succ x^{(2)}\\ -1 & x^{(2)} \succ x^{(1)}\end{cases}$$

Ranking SVM is employed in our personalization approach to learn the user's preferences. For a given query, a set of content concepts and a set of location concepts are extracted from the search result as the document features. Since each document can be represented by a feature vector, it can be treated as a point in the feature 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. It outputs a content weight vector $\overrightarrow{w_{C,q,u}}$ and a location weight vector $\overrightarrow{w_{L,q,u}}$, which best describes the user interests based on the user's content and location preferences extracted from the user clickthroughs, respectively. In the following, we discuss 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.

2) Mining Features for Training: Two feature vectors, namely, content feature vector (denoted by $\phi_C=(q,d)$) and location feature vector (denoted by noted by $\phi_L=(q,d)$ are defined 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 content concepts are 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:

$$\forall c_i \in s_k, \phi_C(q,d_k)[c_i]=\phi_C(q,d_k)[c_i]+1 \quad (5)$$

For other content concepts c_i that are related to the content concept c_i (either they are similar or c_j is the ancestor / descendant /sibling of c_i) in the content ontology, they are incremented in the content feature vector $\phi_C=(q,d_k)$ according to the following equation:

$$\forall c_i \in s_k; \phi_C(q,d_k)[c_j]=\phi_C(q,d_k)[c_j]+sim_R(c_i,c_j)+ ancestor(c_i,c_j)+descendant(c_i,c_j)+sibling(c_i,c_j) \quad (6)$$

Location Feature Vector

If location concepts are l_i is in a web-snippet d_k, their values are incremented in the location feature vector $\phi_L=(q,d_k)$ with the following equation:

$$\forall l_i \in d_k, \phi_L(q,d_k)[l_i]=\phi_L(q,d_k)[l_i]+1 \quad (7)$$

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

$$\forall l_i \in d_i; \phi_L(q,d_k)[l_j]=\phi_L(q,d_k)[c_j]+ancestor(l_i,l_j)+ descendant(l_i,l_j)+sibling(l_i,l_j) \quad (8)$$

3) Combining Weight Vectors: The content feature vector $\phi_C=(q,d)$ together with the document preferences obtained from SpyNB methods are served as input to RSVM training to obtain the content weight vector $\overrightarrow{w_{C,q,u}}$. The location weight vector $\overrightarrow{w_{L,q,u}}$ is obtained similarly using the location feature vector $\phi_L=(q,d)$ and the document preferences. The two weights vectors $\overrightarrow{w_{C,q,u}}$ and $\overrightarrow{w_{L,q,u}}$ represent the content and location user profiles for a user u on a query q in our *ontology-based, multi-facet (OMF)* user profiling method.

To optimize the personalization effect, we use the following formula to combine the two weight vectors, $\overrightarrow{w_{C,q,u}}$ and $\overrightarrow{w_{L,q,u}}$, linearly according to the values of the personalization effectiveness parameters, $e_C(q,u)$

and $e_L(q)$, to obtain the final weight vector $w_{q,u}$ for user u's ranking. The two weight vectors, $\overrightarrow{w_{C,q,u}}$ and $\overrightarrow{w_{L,q,u}}$, are first normalized before the combination.

$$w_{q,u} = \frac{e_C(q,u).\overrightarrow{w_{C,q,u}}}{e_C(q,u)+e_L(q,u)} + \frac{e_L(q,u).\overrightarrow{w_{L,q,u}}}{e_L(q,u)+e_L(q,u)} \quad (9)$$

Let $e(q,u)=\dfrac{e_C(q,u)}{e_C(q,u)+e_L(q,u)}$, then we get the following formula

$$\overrightarrow{w_{q,u}} = e(q,u).\overrightarrow{w_{C,q,u}} + (1-e(q,u)).\overrightarrow{w_{L,q,u}} \quad (10)$$

After the final weight vector, $\overrightarrow{w_{q,u}}$, is computed, a linear ranking function is adopted for rank adaptation of future search results. The documents in the future search will be ranked according to the following formula.

$$f(q,d) = \overrightarrow{w_{q,u}}.\phi(q,d) \quad (11)$$

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.

*4) **User Profile**:* We considered three factors such as content concepts, location concepts and time preference in which exploiting timing enables us to capture the shifts of user's interests based on the time of the day and adapt his preferences accordingly. It provides a means to effectively merge user's preferences under the appropriate time zone which creates a dynamic user's profile. This dynamic profile can accurately cover the preference of a user at all times and situations. While creating and updating user profile according to content and location concepts which are associated with the query and user preferences on both concepts for that query over a time increases the effectiveness rate of web search according to user interest.

3. EXPERIMENTAL RESULTS

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

20 users are invited to submit totally 100 test queries to our search engine. Totally 100 test queries consists 10 categories. Each of the 20 users is assigned 5 test queries randomly selected from the 10 different categories. The users are given the tasks to find results that are relevant to their interests. The clicked results are stored in the click through database along with the time zones and are treated as positive samples in RSVM training. The search engine performs personalized ranking of the search results based on the learnt profiles of the users.

For testing, again the same 20 users are invited. 5 queries are allocated for each user. The queries are randomly selected from 10 different categories. Table 3 shows the performance rate of the search engine regarding user 1.

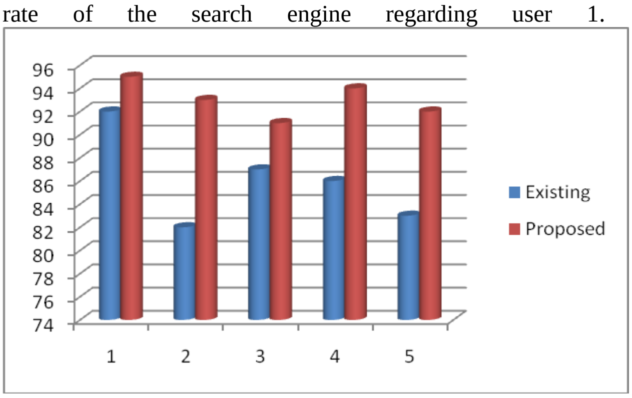

Fig 1: Performance rate of a proposed time based search for user 1

Table 2 Statistics to display the relevant links within top 20 links

User 1						
	Existing			Proposed		
Query	No of Results	Irrevalent result	Accuracy	No of Results	Irrevalent result	Accuracy
1	76	6	92	75	3	95
2	56	10	82	56	4	93
3	45	6	87	42	4	91
4	50	7	86	50	3	94
5	60	10	83	60	5	92

Table 3 Average Performance rate

Uesr	Existing	Proposed
U1	86	93
U2	82	91
U3	83	90
U4	79	89
U5	82	92
U6	80	89
U7	85	92
U8	75	89
U9	78	91
U10	79	93
U11	81	94
U12	82	92
U13	80	94
U14	84	93
U15	79	89
U16	75	88
U17	74	89
U18	73	83
U19	72	89
U20	75	90

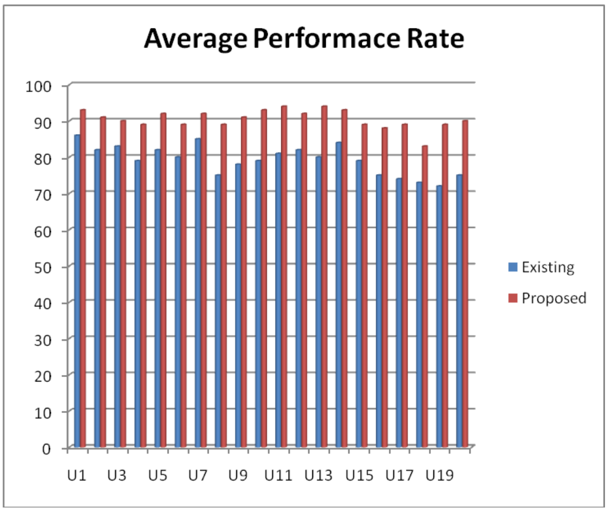

Fig 2: Average performance rate of a proposed time based search

Table 3 and Fig. 2 show that personalizing web search based on content, location and time preferences provide high performance rate in terms of getting relevant information based on user's current interest or preferences. The performance rate provided by the proposed personalized web search is higher than the existing one.

4. CONCLUSION

In this thesis, we have studied some problems in real world information access from the web through web search engines. In this dissertation, we used ontology for web user personalization and search with content, location and time preferences. In this we are extracting the concepts which are related to the query. These are divided into content and location concepts which are useful to create two ontologies such as content and location. Furthermore, we have proposed an idea that, user preferences according to the time zones are extracted which will create click through ontology based on time zones. Based on the ontologies user profile will be created and updated using RSVM.

The experimental result shows that our proposed personalization approach provides higher performance rate compare with conventional methods.

5. SCOPE FOR FUTURE WORK

In recent years, personalized search has attracted interest in the research community as a means to decrease search ambiguity and return results that are more likely to be interesting to a particular user and thus providing more effective and efficient information access. In this dissertation, we have proposed web user personalization and search with content, location and time preference of a user which helps user to get highly relevant information according his/her current interest.

In future, in order to increase the effectiveness of the personalization of web search we are extracting the physical location of a user and for the location based query the search engine will react based on the physical location.

6. BIBLIOGRAPHY

[1] How Stuff Works "How Internet Search Engines Work"

computer.howstuffworks.com/internet/basics/search-engine.htm

[2] Kenneth Wai-Ting Leung, Dik Lun Lee, Wang-Chien Lee

"Personalized Web Search with Location Preferences" ICDE Conference 2010.

[3] K. W. Church, W. Gale, P. Hanks, and D. Hindle, Using statistics in lexical analysis,. Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon, 1991.

[4] National geospatial. http://earth-info.nga.mil/.

[5] World gazetteer. http://www.world-gazetteer.com/.

[6] C. E. Shannon "Prediction and entropy of printed English", Bell Systems Technical Journal, pp. 50.64, 1951.

[7] Christoforos Panayiotou, Maria Andreou, George Samaras, Andreas Pitsillides "Time Based Personalization for the Moving User" Proceedings of the International Conference on Mobile Business (ICMB'05)

[8] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web", Proc. Ninth IEEE Int'l Conf. Tools with AI (ICTAI '97), pp. 558-567, 1997.

[9] O. Nasraoui, R. Krishnapuram, and A. Joshi, "Mining Web Access Logs Using a Relational Clustering Algorithm Based on a Robust Estimator," Proc. Eighth

Int'l World Wide Web Conf. (WWW '99), pp. 40-41, 1999.

[10] O. Nasraoui, R. Krishnapuram, H. Frigui, and A. Joshi, "Extracting Web User Profiles Using Relational Competitive Fuzzy Clustering," Int'l J. Artificial Intelligence Tools, vol. 9, no. 4, pp. 509-526, 2000.

[11] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data," SIGKDD Explorations, vol. 1, no. 2, pp. 1-12, Jan. 2000.

[12] M. Spiliopoulou and L.C. Faulstich, "WUM: A Web Utilization Miner," Proc. First Int'l Workshop Web and Databases (WebDB '98), 1998.

[13] T. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal, "From User Access Patterns to Dynamic Hypertext Linking," Proc. Fifth Int'l World Wide Web Conf. (WWW '96), 1996

[14] M. Perkowitz and O. Etzioni, "Adaptive Web Sites: Automatically Learning for User Access Pattern," Proc. Sixth Int'l WWW Conf. (WWW '97), 1997.

[15] J. Borges and M. Levene, "Data Mining of User Navigation Patterns," Web Usage Analysis and User Profiling, LNCS, H.A. Abbass, R.A. Sarker, and C.S. Newton, eds. pp. 92-111,Springer-Verlag, 1999.

[16] O. Zaiane, M. Xin, and J. Han, "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs," Proc. Advances in Digital Libraries (ADL '98), pp. 19-29, 1998.

[17] O. Nasraoui and R. Krishnapuram, "A New Evolutionary Approach to Web Usage and Context Sensitive Associations Mining," Int'l J. Computational Intelligence and Applications, special issue on Internet intelligent systems, vol. 2, no. 3, pp. 339-348, Sept. 2002.

[18] O. Nasraoui, C. Cardona, C. Rojas, and F. Gonzalez, "Mining Evolving User Profiles in Noisy Web Clickstream Data with a Scalable Immune System Clustering Algorithm," Proc. Workshop Web Mining as a Premise to Effective and Intelligent Web Applications (WebKDD '03), pp. 71-81, Aug. 2003.

[19] P. Desikan and J. Srivastava, "Mining Temporally Evolving Graphs," Proc. Workshop Web Mining and Web Usage Analysis (WebKDD' 04), 2004.

[20] O. Nasraoui, C. Rojas, and C. Cardona, "A Framework for Mining Evolving Trends in Web Data Streams Using Dynamic Learning and Retrospective Validation," Computer Networks, special issue on Web dynamics, vol. 50, no. 14, Oct. 2006.

[21] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent", in Proc.of ICML Conference, 2005.

[22] Q. Gan, J. Attenberg, A. Markowetz, and T. Suel "Analysis of geographic queries in a search engine log", in Proc. of the International Workshop on Location and the Web, 2008.

[23] T. Joachims "Optimizing search engines using click through data", in Proc. of ACM SIGKDD Conference, 2002.

[24] K. W.-T. Leung, W. Ng, and D. L. Lee "Personalized concept-based clustering of search engine queries", IEEE TKDE, vol. 20, no. 11, 2008.

[25] B. Liu, W. S. Lee, P. S. Yu, and X. Li "Partially supervised classification of text documents, in Proc. of ICML Conference, 2002.

[26] W. Ng, L. Deng, and D. L. Lee, "Mining user preference using spy voting for search engine personalization", ACM TOIT, vol. 7, no. 4, 2007.

[27] Q. Tan, X. Chai, W. Ng, and D. Lee "Applying co-training to click through data for search engine adaptation", in Proc. of DASFAA Conference, 2004.

[28] S. Yokoji "Kokono search: A location based search engine", in Proc. of WWW Conference, 2001.

[29] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma "Hybrid index structures for location-based web search", in *Proc. of CIKM Conference*, 2005.

[30] Christoforos Panayiotis, Maria Andreou, George Samaras and Andreas Pitsillides "Time Based Personalization for the Moving User" Proceedings of the International Conference on Mobile Business (ICMB'05), 2005.

[31] Ee-Peng Lim and Aixin Sun "Web Mining – The Ontology approach"

[32] E. Agichtein, E. Brill, and S. Dumais "Improving web search ranking by incorporating user behavior information", in *Proc. of ACM SIGIR Conference*, 2006.