

User Preference Retrieval using Semantic Categorization for Web Search

Teerakorn Bounoy, Aranya Walairacht

*Department of Computer Engineering,
King Mongkut's Institute of Technology Ladkrabang, Thailand
kornnisterrooy@hotmail.com, kwaranya@kmitl.ac.th*

Abstract — Search engines have been one of the most popular ways for people to find web pages of interest. Presently, when a user enters a keyword in a search engine, the search results are usually presented the same result to other users who search the same keyword, which might not be related to each user's field of interest. Therefore, the researcher of this study would like to propose a new searching technique to get each user's most relevant information by using a user preference. This research will categorize user preference to build the user profile and general profile base on user's search history and category hierarchy, respectively. The search engines then use those profiles to determine the interests of each user, execute the search query to obtain a set of relevant documents, and re-ranking the documents in a manner that best reflects their relevance to the user's profile. Many algorithms have been designed, analyzed, implemented and experimented to find the most appropriate and the most effective one to create the relationship between each keyword and each category to best meet each user's preference.

Keywords — Personalization, Clickthrough data, Web search, User preferences, Categorization.

I. INTRODUCTION

As the amount of information on the Web rapidly increases, it creates many new challenges for Web search. When the same query is submitted by different users, a typical search engine returns the same result, regardless of who submitted the query. This may not be suitable for users with different information needs. For example, for the query "penguin," some users may be interested in documents dealing with "penguin" as "Animal," while some other users may want documents related to penguin computers. One way to disambiguate the words in a query is to associate a small set of categories with the query. For example, if the category "Linux" or the category "operating system" is associated with the query "penguin," then the user's intention becomes clear. Current search engines such as Google or Yahoo! have hierarchies of categories to help users to specify their intentions. The use of hierarchical categories such as the Library of Congress Classification is also common among librarians.

A user may associate one or more categories to his/her query manually. For example, a user may first browse a hierarchy of categories and select one or more categories in the hierarchy before submitting his/her query. By utilizing the selected categories as a context for the query, a search engine is likely to return documents that are more suitable to the user. Unfortunately, a category hierarchy shown to a user is usually very large and, as a result, an ordinary user may have difficulty in finding the proper paths leading to the suitable categories. Furthermore, users are often too impatient to identify the proper categories before submitting their queries. An alternative way to browse is to obtain a set of categories for a user query directly by a search engine. However, categories returned from a typical search engine are still independent and base on particular user and many of those returned categories do not reflect the intention of the searcher.

To solve these problems, we propose a two-step strategy to improve retrieval effectiveness. In the first step, the system automatically deduces, for each user, a small set of categories for each query submitted by the user, based on his/her search history. In the second step, the system uses the set of categories to augment the query to conduct the Web search. Specifically, we provide a strategy to:

1. model and gather the user's search history,
2. construct a user profile based on the search history and construct a general profile based on the ODP (Open Directory Project) category hierarchy,
3. deduce appropriate categories for each user query based on the user's profile and the general profile,
4. improve Web search effectiveness by using these categories as a context for each query.

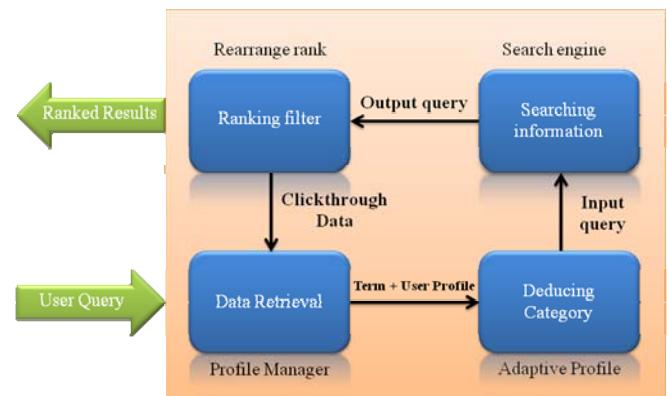


Figure 1. The basic process of search engine adaptation

Numerous experiments are performed to demonstrate that our strategy of personalized Web search is both effective and efficient.

The basic process of search engine adaptation in our new approach is shown in Figure 1.

The rest of this article is organized as follows: Section 2 surveys the related work. In Section 3, propose user profile and general profile. In Section 4, we report empirical results when user profile and general profile are applied to web search. Finally, Section 5 concludes the article.

II. RELATE WORK

Many techniques are used in modern search engines to provide the results for user queries. Yahoo! (<http://www.yahoo.com/>), ODP (<http://dmoz.org/>) and Google (<http://www.google.com/>) return both categories and documents. Northern Light (<http://www.northernlight.com/>) and WiseNut (<http://www.wisenut.com/>) cluster their results into categories and Vivisimo (<http://www.vivisimo.com/>) groups results dynamically into clusters. Teoma (<http://www.teoma.com/>) clusters its results and provides query refinements. A lot of research in metasearch and distributed retrieval [5], [6] also investigates mapping user queries to a set of categories or collections. However, all of the above techniques return the same results for a given query, regardless of who submitted the query. This can be interpreted as having a general profile. The results of experimental in this paper indicate that using the combination of a user profile and a general profile usually yields significantly higher accuracy than using a general profile or a user profile.

The investigation about user preferences [9] provide category for searching, but it does not cover enough, while [10] assign users to select the category. For example, a user may first browse a hierarchy of categories and select one or more categories in the hierarchy before submitting his/her query. By utilizing the selected categories as a context for the query, a search engine is likely to return documents that are more suitable to the user. Unfortunately, a category hierarchy shown to a user is usually very large and, as a result, an ordinary user may have difficulty in finding the proper paths leading to the suitable categories. Furthermore, users are often too impatient to identify the proper categories before submitting their queries. [4], [7], [8], [11], [12] investigate only data which user click on, while this paper consider the data which user click and don't click on. Because some results which user don't select may have significant for some reasons (see section 3A).

Many papers on information filtering [7], [9] also construct user profiles explicitly or implicitly and recommend documents using the profiles and filter information using documents. While this paper determine the categories which are likely to be the intention of the user. The determined categories are used as a context for the user query to improve retrieval effectiveness. Furthermore, no general profile was used in information filtering in previous papers.

In the area of personalized Web search [4], [8] manage categories by using ODP database. Those papers define levels

of categorize only three level. And the experimental does not show results when users change their preferences. While this paper does not concern about level of categorize and show the results after user change their query to another. [9] uses user profiles to refine user queries, but does not improve the user profile when user select exist documents, while this paper always improve the user profile by refer previous weight of user's search history.

III. PROFILE MANAGEMENT

This approach propose new clickthrough interpretation by considering link that user does not clicked, also design user profile and general profile for mapping term and category which the result will be used for choosing category for next time's searching.

A. Clickthrough Interpretation

In Table 1, after analyzing the titles, abstracts, and URLs of all 6 links, found that the links are about two different topics: links l_2 , l_3 , l_4 and l_5 are about "penguin Linux" while the other 2 links are related to "penguin animal".

Table 1. Search on the query *penguin* and the Clickthrough data

Link	The lists of search results.
l_1	Penguins ..., scientific classification, distribution and habitat, physical characteristics, for further research. http://www.seaworld.org/animal-info/info-books/penguin/
l_2	Linux Penguins on the web A collection of links to Tux sites. http://tunes.org/~do/penguinindex.html
l_3 (clicked)	Temple of the Screaming Penguin Linux-based site. Content is fairly technical; site has many links to other sites and articles... http://www.screaming-penguin.com/
l_4	Tux Reports: Where Penguins Fly Focusing on GNU/Linux, Tux Reports provides reviews, commentary, forum discussions... http://www.tuxreports.com/
l_5 (clicked)	Place for Penguin Lovers Linux news, FAQs, and links. http://jonid.8m.com/Linux/Index.htm
l_6	Pete and Barb's Penguin Pages Extensive site with photos, maps, accounts for all species, FAQ, behavior, and where to see penguins... http://www.adelie.pwp.blueyonder.co.uk/

Furthermore, the clicked links l_3 and l_5 (in bold) are all about "penguin Linux" Therefore, an intuitive interpretation of this clickthrough data is that the user is looking for results about "penguin Linux" from a preference mining point of view, we can infer that *the user likes links about "penguin Linux" more than links about "penguin animal"*.

In the above example, the problem of the above interpretation is that they mistakenly identify some high-ranking unclicked links about "penguin Linux" (e.g., l_2 , l_4) as "unpreferred" links. We argue that in practice it is possible that the user does not click on all of the links relevant to his or her interests, because he or she may not be patient enough to examine all the relevant links, or he or she may stop clicking

after seeing “enough” information, thus leave some relevant links unclicked. Moreover, a user may skip a relevant link because the abstract of that link is not informative enough. However, above interpretation cannot handle the above-mentioned possibilities but simply derive preferences based on the simple rule that *if a high-ranking link is not clicked, it is then considered an “unpreferred” link*.

Since this approach uses category to indicate user preferences and each link has specific category, so it is easy to analyze and group the clickthrough data by using new interpretation.

B. Profile Structure

Figure 2. show the user’s search history model.

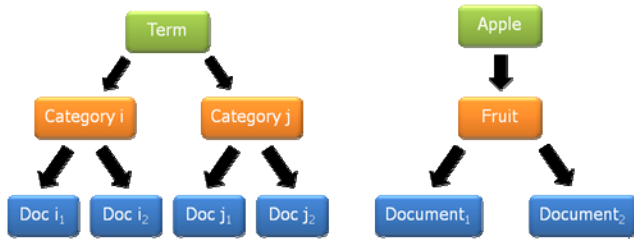


Figure 2. A model and an example of a search record.

The root of a search record is a query. Each query has one or more related categories. Associated with each category is a set of documents, each of which is both relevant to the query and related to the category. So the profile structure consists of three elements:

1) **Term:** query or data which user need to search. For example, penguin or apple.

2) **Category:** classification or group of document which relate to term. For example the term penguin is related to both Linux and Animal. So Linux and Animal are the categories of the search.

3) **Document:** details of each link which relate to term.

In this paper, there are two types of profile:

1) **User profile:** represent users’ interests (user’s search history) and to infer their intentions for new queries.

2) **General profile:** The general knowledge used here which is applicable to all users.

These profiles hold the same structure. User profile can be updated depends on user’s search history while general profile cannot be updated because it represents knowledge of all users.

C. User Profile

User profile represents user’s search history. In this paper user profile consist of set of category, keywords which relate to that category and weight of each keyword. Each weight shows interest of user in each category. For example if weight of keyword “penguin” in category “Linux” has more value than category “Animal” when user search next time with keyword “penguin”, the result should show document about

“penguin Linux” before “penguin animal”. Weight in user profile can compute from three matrices:

- Document - Term Matrix
- Document - Category Matrix
- Category - Term Matrix

D. Document-term matrix (DTM)

Matrix DT ($m \times n$) is a document-term matrix, which is constructed from the user queries and the relevant documents. m is the number of documents in a user’s search history and n is the number of distinct terms occurring in these documents. Each query or relevant document is a row vector of weighted terms in DT. If a term, say term j , occurs in the i^{th} query/relevant document, the weight $DT(i, j) > 0$ otherwise it is 0. The value of $DT(i, j)$ is determined by the average of common normalized tf-idf and LSI weight scheme.

Table 2. Document – Term Matrix of user profile

Document / Term	penguin	Linux	species	tux
D ₁ (Link 1)	1.00	0.00	0.00	0.00
D ₂ (Link 2)	0.67	0.52	0.00	0.00
D ₃ (Link 3)	0.00	0.00	1.00	1.00
D ₄ (Link 4)	0.00	0.00	0.52	0.52

The above Table 2 is a sample to show the structure of DTM. The values of the cells are calculated by average tf-idf and LSI Scheme which is explained as follows:

1) **Term Frequency-Inverse Document Frequency (tf-idf):** The tf-idf weight (term frequency–inverse document frequency) is a weight often used in information retrieval and text mining. The *term frequency* (tf), defined as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ is the number of occurrences of the considered term (t_i) in document d_j , and the denominator is the sum of number of occurrences of all terms in document d_j .

The *inverse document frequency* (idf) is a measure of the general importance of the term.

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

where $|D|$: total number of documents in the corpus.
 $|\{d : t_i \in d\}|$: number of documents where the term t_i appears.
 Then

$$(tf-idf)_{i,j} = tf_{i,j} \times idf_i$$

For example, considering a document containing 100 words where the word *penguin* appears 3 times. Following the previously defined formulas, the term frequency (tf) for *penguin* is then 0.03 (3/100). Now, assume we have 10 million documents and *penguin* appears in one thousand of these. Then, the inverse document frequency is calculated as

$\ln(10,000,000/1,000) = 9.21$. The tf-idf score is the product of these quantities: $0.03 * 9.21 = 0.28$.

2) Latent Semantic Indexing (LSI): LSI is based on the principle that words are used in the same contexts tend to have similar meanings. For example drink, drinks, drinking, drinker and drinkers. The LSI defined as follows:

$$g_i = 1 + \sum_j (p_{ij} \log(p_{ij}) / \log(n))$$

which $p_{ij} = tf_{ij} / gf_i$

where gf_i is the total number of times term i occurs in the whole collection and tf_{ij} is the number of occurrences of term i in document j . n is total documents. Then

$$a_{ij} = g_i \log(tf_{ij} + 1)$$

where g_i describes the relative frequency of the term within the entire collection of documents.

E. Document - Category Matrix (DCM)

For each row in matrix DT ($m * p$), there is a corresponding row in the matrix DC. The columns (p) of DC are the set of related categories. Since a row in DT represents a query/document, the corresponding row in the matrix DC indicates the set of categories related to the query/document. More precisely, if there is an edge between the j^{th} category and the i^{th} query/document in the tree model of a search record, then the entry DC (i, j) = 1; otherwise it is 0. The Document Category Matrix corresponding to Document Term Matrix presented earlier is as follows.

Table 3. Document – Category Matrix of user profile

Document / Category	Linux	Animal
D ₁ (Link 1)	0	1
D ₂ (Link 2)	1	0
D ₃ (Link 3)	1	0
D ₄ (Link 4)	1	0

F. Category - Term matrix (CTM)

From DTM and DCM, we learn a matrix CTM ($p * n$), which represents the user profile. Each row in the matrix CT which represents a category of interest to the user, is a vector of weighted terms. Thus, both categories and documents are represented in the same vector space of terms and similarities between them can be computed show in table 4. The weight of each cells are calculated by Rocchio – Based Algorithm follow:

Table 4. Category - Term Matrix of user profile

Category / Term	penguin	Linux	species	tux
Linux	0.83	0.47	0.00	0.00
Animal	0.79	0.00	0.68	0.72

1) Rocchio – Based Algorithm: Rocchio [12] is method to compute weight of user profile from relation between DC and DT, after that [11] improved previous algorithm defined as follows:

$$M(i, j)^t = \frac{N_i^{t-1}}{N_i^t} M(i, j)^{t-1} + \frac{1}{N_i^t} \sum_k DT(k, j) \times DC(k, i)$$

where M^t is the modified user profile at time t , N_i^t is the number of documents which are related to the i^{th} category and have been accumulated from time zero to time t , m is the number of documents in DT, $DT(k, j)$ is the weight of the j^{th} term in the k^{th} document, $DC(k, i)$ is a binary value denoting whether the k^{th} document is related to the i^{th} category.

G. General Profile

The general knowledge used here which is applicable to all users is called general profile which is constructed before user's interests are captured, the browser has to present the user, a list of contexts in which the user's search term can be referred. Then from the user's selection the user interest is obtained.

The general knowledge that utilizes is extracted from ODP. The general profile is constructed using the learning Rocchio – Based Algorithm algorithm presented. Let the three corresponding matrixes related to the general knowledge be denoted by DTg, DCg and CTg (g means general profile). Weight of CTg has the highest average accuracy and, although it is computationally expensive, CTg needs to be computed only once.

Table 5. Category - Term Matrix of general profile

Category / Term	penguin	Linux	species	tux
Linux	1.97	1.31	0.00	0.00
Animal	1.33	0.00	1.27	1.22

IV. EXPERIMENTAL

In this paper, we conducted experiments by using the ODP database (update 06-Nov-2009 06:06) to evaluate the effectiveness of user preference technique. The accuracy [12] was used to compute and compare this approach with the common way of choosing documents.

A. Experimental Setup

These experiments use three groups of employees from three different departments at the S-One telecom company, namely, Technical, Account, and Production, to use the search engine. Each group had 5 people. These experiments assumed the following about these subjects: users from different departments had different interests but users within the same department shared the same interests. The people from Technical were looking for computer information; the people from Account were interested in finances; and the people from Production preferred to receive news and modern products. As far as the personalization method is concerned, the three groups of employees could be considered as three “logical” users and the personalization method tried to adapt the search engine to deliver the best results to the respective group of users. Using more than one employee in each group ensured that the experimental results were not affected by a few peculiar actions made by one or two users.

To collect the clickthrough data, each member of the three groups of employees submitted to the search engine 30 queries related to his or her interests. Thus there were 150 query submissions in total from each group. The search engine at the beginning adopted the general profile to deliver results. After users click on the link, the user profile has been improved and delivers result to each user. Table 6 shows some statistics of the clickthrough data which collected in this paper.

Table 6. Statistics of Clickthrough Data Sets

Statistics	Technical	Account	Production
# of submitted queries	150	150	150
# of clicks	472	533	496
# of interest categories	24	29	43
# of distinct terms	10412	13137	10975
Avg. clicks per category	19.67	18.38	11.53
Avg. clicks per query	3.15	3.55	3.30
Avg. rank clicked on	1.54	2.77	1.78

B. Performance Measures

1) Accuracy of Mapping User Queries to Categories: In this approach, the category levels are returned for each user query. The following performance metric is proposed:

$$\text{Accuracy} = \left(\sum score_{ci} \right) / n$$

$$= \left(\sum \frac{1}{1 + rank_{ci} - ideal_rank_{ci}} \right) / n$$

where n is the number of related categories to the query, $score_{ci}$ is the score of a related category c_i , $rank_{ci}$ is the rank of c_i , and $ideal_rank_{ci}$ is the highest possible rank for c_i . The accuracy was computed for each query. For example, assume that c_1 and c_2 are related categories to a user query and they are ranked by the system to be the second and the fifth, then the accuracy should be computed in the following way: $score_{c1} = 1/(1+2-1) = 0.5$ and $score_{c2} = 1/(1+5-4) = 0.5$ so the accuracy is $(0.5+0.5)/2 = 0.5$

C. Experimental Results

1) Results of Mapping User Queries to Categories: In first result, this paper investigate the effectiveness of two algorithms to calculate weight of CT which are: tf-idf, LSI algorithm include average of tf-idf and LSI. Table 7 shows their accuracy results.

Table 7. Result mapping using tf-idf, LSI and average

Method	Weight
tf-idf	0.8234
LSI	0.7340
Avg.(tf-idf+LSI)	0.7787

Secondly, examining the effects of combining the user profile and the general profile, and comparing the three combining methods with the two baselines (one using the general profile only and another using the user profile only).

The general profile was constructed by using Rocchio – Based with average tf-idf and LSI to compute CT. Figure 2 show the results of comparison.

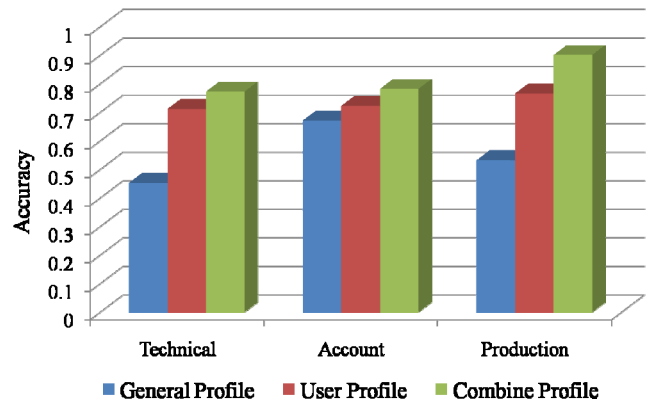


Figure 2. Result of comparison in different method

2) Evaluation of Clickthrough Data Quality: In order to compare the effectiveness of learning profile and display result between links which users click with entire links (click and not click), this approach uses average tf-idf and LSI and Rocchio – Based to learn profile. The result of rank of users' clicks show in Figure 3.

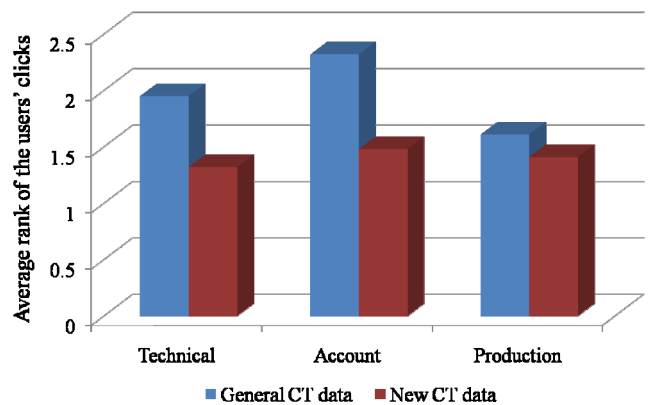
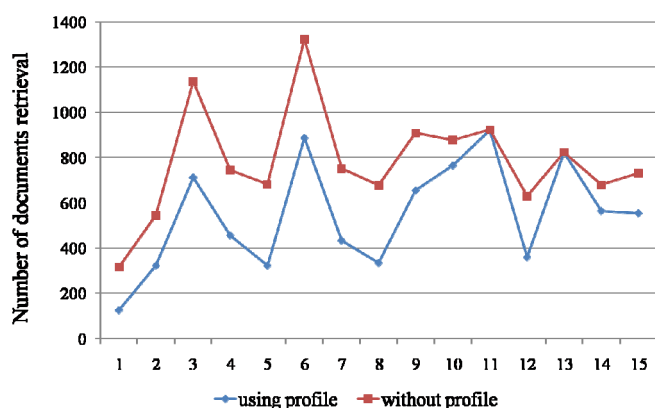


Figure 3. Rank of users' clicks

The result in Figure 3 shows that, computing the links that users don't click give higher rank than links that the users want. Thus, the smaller the average rank of the users' clicks, the better the ranking quality. The values of average rank using entire links all about 1.7, which means that the learning profile using entire links satisfies the three user groups better than the learning profile using only links that users clicked.

3) Reduction in records retrieval by the search engine: The documents retrieved by the search engine using profiles show in Figure 4. The results show that the record's number of retrieved documents using profiles less than using search without profiles.



Example of query times in Technical department

Figure 4. Retrieval documents using profile and without profiles

V. CONCLUSION

A technique to learn user profiles from users' search histories which can be collected without direct user involvement has been proposed in this paper. The user profiles can be constructed automatically from the user's search history and is augmented by a general profile which is extracted automatically from a common category hierarchy and used to improve retrieval effectiveness in Web search based on a new clickthrough interpretation. The categories that are likely to be of interest to the user are deduced based on his/her query and the two profiles. The experimental results show the accuracy of using both profiles is consistently better than using only the user profile or only general profile and the accuracy of using average tf-idf and LIS algorithm to compute DTM is more efficiency than using only one of them. The result of CTM's weights base on new clickthrough interpretation which shows three groups of users satisfy this method more than previous method that investigate only link which user click on. And the results also indicate that using of this technique reduces the number of documents retrieved by the search engine is reduced to 30%.

ACKNOWLEDGMENT

The author would like to thank Asst.Prof.Dr.Aranya Walairacht, Kairika Srithep and Themjai Srithep, the employees of S-One telecom, and the anonymous reviewer for useful comments and suggestions.

REFERENCES

- [1] Welfred Ng, Lin Deng, and Dik Lun Lee, "Mining User Preference Using Spy Voting for Search Engine Personalization", ACM Transactions on Internet Technology, Vol. 7, No. 4, Article 19, Publication date: October 2007, pp. 19-45.
- [2] Joachims, T. 2002b, "Optimizing search engines using clickthrough data", In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (KDD 02), pp. 133-142.
- [3] Joachims, T., Granka L. A., Pan, B., Hembrooke, H., and Gay, G. 2005. "Accurately interpreting clickthrough data as implicit feedback", In Proceedings of SIGIR, pp.154-161.
- [4] Fang Liu, Clement Yu and Weiyi Meng, "Personalized Web Search for Improving Retrieval Effectiveness", IEEE TRANSACTIONS ON

KNOWLEDGE AND DATA ENGINEERING, Vol. 16, No. 1, January 2004, p28-40.

- [5] R. Dolin, D. Agrawal, A. El Abbadi, and J. Pearlman, "Using Automated Classification for Summarizing and Selecting Heterogeneous Information Sources", D-Lib Magazine, 1998.
- [6] C.T. Yu, W. Meng, W. Wu, and K.-L. Liu, "Efficient and Effective Metasearch for Text Databases Incorporating Linkages among Documents", ACM SIGMOD, pp. 187-198, 2001.
- [7] J. Allan, "Incremental Relevance Feedback for Information Filtering", Proc. 19th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 270-278, 1996.
- [8] John Garofalakis, Theodoula Giannakoudi, and Agoritsa Vopi, "Personalized Web Search by Constructing Semantic Clusters of User Profiles", I. Lovrek, R.J. Howlett, and L.C. Jain (Eds.): KES 2008, Part II, LNAI 5178, pp. 238-247, 2008.
- [9] M.R. Sumalathal, V. Vaidehi, A. Kannan³ and S. Anandhi, "Information Retrieval using Semantic Web Browser - Personalized and Categorical Web Search", IEEE - ICSCN 2007, MIT Campus, Anna University, Chennai, India. Feb. 22-24, 2007. pp.238-243.
- [10] J. Rocchio, "Relevance Feedback in Information Retrieval", The Smart Retrieval System: Experiments in Automatic Document Processing, pp. 313-323, 1971.
- [11] Fang Liu, Clement Yu, and Weiyi Meng, "Personalized Web Search for Improving Retrieval Effectiveness", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 1, JANUARY 2004, pp. 28-40.
- [12] M.R. Sumalathal, V. Vaidehi, A. Kannan and S. Anandhi, "Information Retrieval using Semantic Web Browser - Personalized and Categorical Web Search", IEEE - ICSCN 2007, MIT Campus, Anna University, Chennai, India. Feb. 22-24, 2007, pp.238-243.