Concept-Based user Profiles for Effective Search

D. Pavithra,
Assistant Professor, CSE
Dr.N.G.P Institute of Technology,
Coimbatore.

***Abstract*— The objective of this project is to create a concept–based user profile using SpyNB-C algorithm. The relationship between the users are mined from the concept-based user profiles to perform collaborative filtering which allow the users with the same interest to share their profiles. The concept-based user profile is integrated into ranking algorithm so that the results are ranked according to the individual user's interest. Concept-based methods automatically derive users' topical interest by exploring the contents of the users' browsed documents and search histories. The user profile is represented as a set of categories, and for each category, a set of keywords with weights. The SpyNB-C identifies the user preference pairs generated from clickthrough data based on the concept. The SpyNB-C algorithm treats clicked pages as positive samples and unclicked pages as unlabeled samples in the training process. Collaborative filtering (CF) is the process of filtering information for a user, based on a collection of user profiles. Users having similar profiles may share similar interests. For a user, information can be filtered in/out regarding to the behaviors of his or her similar users. Ranking is used to display the web results such that the most relevant or authoritative pages are displayed first.**

***Index Terms*—Collaborative Filtering, User Profiling, Negative Preferences.**

I. Introduction

For a given query, a personalized Web search can provide different search results for different users or organize search results differently for each user, based upon their interests, preferences, and information needs. Personalized web search differs from generic web search, which returns identical research results to all users for identical queries, regardless of varied user interests and information needs. To provide personalized search results to users, personalized web search maintains a user profile for each individual. A user profile stores approximations of user tastes, interests and preferences. It is generated and updated by exploiting user-related information. User information can be specified by the user (explicitly collecting) or can be automatically learnt from a user's historical activities (implicitly collecting).

User profiling strategies can be broadly classified into two main approaches: document-based and concept-based approaches. Document-based user profiling methods aim at capturing users' clicking and browsing behaviors. Concept-based user profiling methods aim at capturing users' conceptual needs. Users' browsed documents and search histories are automatically mapped into a set of topical categories. User profiles are created based on the users' preferences on the extracted topical categories.

Collaborative filtering is the method of making automatic predictions (filtering) about the interests of a user by collecting user profile information from many users (collaborating). The similar users' user profiles are shared. Ranking algorithm is based on learning from historical query logs, to predict users' information needs and reduce the seeking time from the search result list.

The main contributions of this paper are:

- First, The concept is extracted by identifying a keyword/phrase that exists frequently in the web-snippets of a particular query. The SpyNB treats clicked pages as positive samples and unclicked pages as unlabeled samples in the training process. The "Spy" technique incorporates a novel voting procedure into Naïve Bayes classifier to derive reliable negative examples from the unlabeled set. The page preferences obtained from SpyNB are generalized into concept preferences[5][2].
- Second, The relationship between the users are identified using Dynamic Agglomerative-Divisive clustering. Dynamic Agglomerative-Divisive Clustering (DADC) algorithm produce clusters of similar users, similar queries and similar concepts based on Community Clickthrough Model (CCM), which replaces clicked documents with concepts embodied in the clicked documents[4].
- Third, A novel ranking method named as QueryFind, based on learning from user profile, is proposed to reduce the seeking time from the search result list. This method uses not only the users' feedback but also the recommendation of a source search engine[1].

II. RELATED WORK

A. *Concept-based user profiling*

Concept-based user profiling methods aim at capturing users' conceptual needs. Users' browsed documents and search histories are automatically mapped into a set of topical categories. User profiles are created based on the users' preferences on the extracted topical categories.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Links	Information of web pages in the search results
l1 clicked	Biometrics Research Page Biometrics.cse.msu.edu
l2	National Cancer Institute – Biometric Research linus.nci.nih.gov/brb
l3	International Biometric Group www.biometricgroup.com
l4	Microsoft Research – Vision Technology research.microsoft.com/research/vision
l5	Forest Biometrics Research Institute www.forestbiometrics.com/Institute.htm
l6	Signal Processing Research Center www.sprc.qut.edu.au/research/fingerprint.html
l7 clicked	Research: Biometrics www.nwfusion.com/research/biometrics.html
l8	Biometrics : Overview biometrics.cse.msn.edu/info.html
l9	TeKey Research Group www.tekey.com
l10 clicked	Biometrics Research Areas IRL www.research.ibm.com/irl/projects/biometrics

Table 1 The clickthrough data for the query "Biometrics Research"

Joachims [8] proposed a method which employs reference mining and machine learning to model users' clicking and browsing behavior. Joachims' method assumes that a user would scan the search result list from top to bottom. If a user has skipped a document d_i at rank i before clicking on document d_j at rank j, it is assumed that he/she must have scan the document d_i and decided to skip it. Thus, we can conclude that the user prefers document d_j more than document d_i (i.e. *dj <r_ di*, where *r_* is the user's preference order of the documents in the search result list).

Preference pairs arising from l_1	Preference pairs arising from l_7	Preference pairs arising from l_{10}
Empty set	$l_7 <_r l_2$	$l_{10} <_r l_2$
	$l_7 <_r l_3$	$l_{10} <_r l_3$
	$l_7 <_r l_4$	$l_{10} <_r l_4$
	$l_7 <_r l_5$	$l_{10} <_r l_5$
	$l_7 <_r l_6$	$l_{10} <_r l_6$
		$l_{10} <_r l_8$
		$l_{10} <_r l_9$

Table 2 Preference pairs derived from clickthrough data using Joachims method

Using Joachims' proposition and the example clickthrough data in Table 1, a set of document preference pairs as shown in Table 2 can be obtained. Ng et al. [9] proposed an algorithm which combines a spying technique together with a novel voting procedure to determine users' document preferences from the clickthrough data. They also employed the RSVM algorithm to learn the user behavior model as a set of weight features.

Gauch et al. [7] proposed a method to create user profiles from user browsed documents. User profiles are created using concepts from the top four levels of the concept hierarchy created by Magellan . A classifier is employed to classify user browsed documents into concepts in the reference ontology.

B. *Query Clustering*

In Beeferman and Berger's agglomerative clustering algorithm [10] (or simply called BB's algorithm in this paper), a query-document bipartite graph is firstly constructed with one set of nodes corresponds to the set of the submitted queries, while the other set of nodes corresponds to the set of clicked documents. When a use submits a query and clicks on a document, the corresponding query and the clicked document are linked together with an edge on the bipartite graph. During the clustering process, the algorithm iteratively merges the two most similar query into one query node, then the two most similar documents into one document node, and the process of alternative merging is repeated until the termination condition is satisfied.

In the web domain, M-LSA [11] represents the relationships between users, queries, and documents with three co-occurrence matrices (*Mu*×*q*, *Mu*×*d*, and *Mq*×*d*), where *u*, *q*, and *d* are the users, queries and documents respectively. A unified co-occurrence matrix *R* is constructed using the co-occurrence matrices. Similar to LSA,M-LSA also employs Eigen Value Decomposition (*EVD*) to discover important objects from the object collections from *R*.

III. USER PROFILING

Our personalized concept-based user profiling method consists of three steps. First, The concept is extracted by identifying a keyword/phrase that exists frequently in the web-snippets of a particular query and The page preferences obtained from SpyNB are generalized into concept preferences. Second, The relationship between the users are identified using Dynamic Agglomerative-Divisive clustering based on Community Clickthrough model. Third, A novel ranking method named as QueryFind, based on learning from user profile, is proposed to reduce the seeking time from the search result list.

A. *Concept Extraction*

We assume that if a keyword/phrase exists frequently in the web-snippets of a particular query, it represents an important concept related to the query because it co-exists in close proximity with the query in the top documents. The following support formula, is used to measure the interestingness of a particular keyword/phrase *ci* extracted from the web-snippets arising from *q*: *interestingness* of a particular keyword/phrase *ci* with respect to the query *q*:

$$support(c_i) = \frac{sf(c_i)}{n} \cdot /c_i/ \qquad (1)$$

Concept c_i	Support(c_i)	Concept c_i	Support(c_i)
Mac	0.1	Apple store	0.06
iphone	0.1	Slashdot apple	0.04
ipod	0.1	Picture	0.04
Hardware	0.09	Music	0.03
Mac os	0.06	Apple farm	0.02

Table 3 Example concept extracted for query "apple"

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

where *sf*(*ci*) is the snippet frequency of the keyword / phrase *ci* (i.e. the number of web-snippets containing *ci*), *n* is the number of web-snippets returned and |*ci*| is the number of terms in the keyword/phrase *ci*. Before concepts are extracted, stopwords, such as "the", "of", "we", etc., are first removed from the snippets[5].

B. SpyNb-C Algorithm

SpyNB treats clicked pages as positive samples and unclicked pages as unlabeled samples in the training process. The problem of finding user preferences becomes one of identifying from the unlabeled set reliable negative documents that are considered irrelevant to the user. The "Spy" technique incorporates a novel voting procedure into a Naïve Bayes classifier to derive reliable negative examples from the unlabeled set. Let "+" and "-" denote the positive and negative classes, and D = d1,d2, ..., dn, a set of N documents in the search result list. For each search result, SpyNB first extracts the words that appear in the title, abstract and URL, creating a word vector (w1,w2, ...,wM) . Then, a Naïve Bayes classifier is built by estimating the prior probabilities (Pr(+) and Pr(−)) and likelihoods (Pr(wj |+) and Pr(wj |−))[2].

The training data only contains positive and unlabeled examples (without negative examples). Thus, the "Spy" technique is employed to learn a Naïve Bayes classifier. A set of positive examples S is selected from P and moved into U as "spies" to train a classifier using the Naïve Bayes algorithm above. The resulting classifier is then used to assign probabilities Pr |+d| to each example in U U S, and an unlabeled example in U is selected as a predicted negative example (PN) if its probability is less than Ts. After obtaining the positive and predicted negative samples from the SpyNB, page preferences can be obtained. As with Joachims-C and mJoachims-C, SpyNB-C generalizes page preferences into concept preferences[2].

C. Dynamic Agglomerative-Divisive Clustering

Dynamic Agglomerative-Divisive Clustering (DADC) algorithm performs updates efficiently by incrementally updating the tripartite graph as new data arrives. It consists of two phases, namely, the agglomerative phase and divisive phase. It prevents clusters from growing without bound when new data arrives. The clickthrough data is first converted into a tripartite graph using Community Clickthrough model (CCM) and DADC would iteratively merge and split nodes in the tripartite graph until the termination condition is reached.

a. Community Clickthrough Model

To alleviate the click sparsity problem, a content-aware clickthrough model, called Community Clickthrough Model (CCM) is introduced, which replaces clicked documents with concepts embodied in the clicked documents. When a user ui submits a query qj , an edge is created between ui and qj representing the relationship between ui and qj . Similarly, if a query qi retrieves a document that embodies concept cj ,an edge is created between qi and cj . When a user ui clicks on a document that embodies concept ck, an edge is created between ui and ck. For clarity, when a user u clicks on a document that embodies a concept c, it is says u clicks on c; if q retrieves a document that embodies concept c, it says u retrieves c. Thus, CCM is a tripartite graph relating users, their submitted queries, the retrieved concepts and the clicked concepts, which are a subset of the retrieved concepts[4].

b. Agglomerative Phase

The agglomerative phase is based on the tripartite graph with the following assumptions: 1) Two users are similar if they submit similar queries and click on similar concepts, 2) Two queries are similar if they are submitted by similar users and retrieve similar concepts, and 3) Two concepts are similar if they are clicked by similar users and are retrieved by similar queries. the following similarity functions are proposed to compute the similarity between pair of users, pair of queries, and pair of concepts.

$$\mathrm{Sim}(u_i, u_j) = \alpha_1 \,.\, \frac{Q_{ui} \,.\, Q_{uj}}{\| Q_{ui} \| \, \| Q_{uj} \|} + \beta_1 \,.\, \frac{C_{ui} \,.\, C_{uj}}{\| C_{ui} \| \, \| C_{uj} \|} \qquad (2)$$

$$\mathrm{Sim}(u_i, u_j) = \alpha_2 \,.\, \frac{U_{qi} \,.\, U_{qj}}{\| U_{qi} \| \, \| U_{qj} \|} + \beta_2 \,.\, \frac{C_{qi} \,.\, C_{qj}}{\| C_{qi} \| \, \| C_{qj} \|} \qquad (3)$$

$$\mathrm{Sim}(u_i, u_j) = \alpha_3 \,.\, \frac{U_{ci} \,.\, U_{cj}}{\| U_{ci} \| \, \| U_{cj} \|} + \beta_3 \,.\, \frac{Q_{ci} \,.\, Q_{cj}}{\| Q_{ci} \| \, \| Q_{cj} \|} \qquad (4)$$

where Q_{ui} is a weight vector for the set of neighbor query nodes of the user node ui in the tripartite graph G3, the weight of a query neighbor node $q_{(k,ui)}$ in the weight vector Q_{ui} is the weight of the link connecting u_i and $q_{(k,qi)}$ in G3. C_{ui} is a weight vector for the set of neighbor concept nodes of the user node u_i in G3, and the weight of a query neighbor node $c_{(k,ui)}$ in C_{ui} is the weight of the link connecting ui and $c_{(k,ui)}$ in G3. Similarly, U_{qi} is a weight vector for the set of neighbor user nodes of the query node q_i, C_{qi} is a weight vector for the set of neighbor concept nodes of the query node q_i, U_{ci} is a weight vector for the set of neighbor user nodes of the concept node c_i,and Q_{cj} is a weight vector for the set of neighbor query nodes of the concept node c_i. In the agglomerative phase, the algorithm merges the two most similar users, then the two most similar queries are merged, and finally the two most similar concepts are merged, and so on. The procedure repeats until no new cluster (user, query or document cluster) can be formed by merging [4].

c. Divisive Phase

The divisive phase employs a hierarchical clustering technique, which is an inverse of the agglomerative phase (splitting instead of merging). It iteratively splits large clusters into two smaller clusters until no new clusters can be formed by splitting. In the divisive phase, each cluster is assigned with

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

a different ε, namely, ε_k. Assume that the distances between pair of users, pair of queries, and pair of concepts are defined as follows.

$$\text{€} = \frac{\sqrt{R^2\ln(1/\delta)}}{2n} \qquad (5)$$

$$d(u_i,u_j) = \sqrt{\sum_{k=1}^{n} (q_{(k,ui)}-q_{(k,uj)})^2+\sum_{k=1}^{n}(c_{(k,ui)}-c_{(k,uj)})^2} \qquad (6)$$

$$d(q_i,q_j) = \sqrt{\sum_{k=1}^{n} (u_{(k,qi)}-u_{(k,qj)})^2+\sum_{k=1}^{n}(c_{(k,qi)}-c_{(k,qj)})^2} \qquad (7)$$

$$d(c_i,c_j) = \sqrt{\sum_{k=1}^{n} (u_{(k,ci)}-u_{(k,cj)})^2+\sum_{k=1}^{n}(q_{(k,ci)}-q_{(k,cj)})^2} \qquad (8\ )$$

$q_{(k,ui)}$ € Q_{ui} is the weight of the link connecting u_i and $q_{(k,ui)}$,and $c_{(k,ui)}$ € C_{ui} is the weight of the link connecting u_i and $c_{(k,ui)}$. Similarly, $u_{(k,qi)}$ € U_{qi} , $c_{(k,qi)}$ € C_{qi} , $u_{(k,ci)}$ € U_{ci} ,and $q_{(k,ci)}$ € Q_{ci} . Assume that two pairs of nodes ($d_{1n} = d_{(ni,nj)}$ and $d_{2n} = d_{(nk,nl)}$) are the topmost and second top-most dissimilar nodes in a cluster. Assume that $\Delta d = d_{(ni,nj\ )} - d_{(nk,nl)}$,if $\Delta d>\varepsilon_k$, with probability $1 - \delta$, the differences between $d_{(ni,nj)}$ and $d_{(nk,nl)}$ is large than zero, and pick (ni,nj) as the boundary of the cluster. In the divisive phase, ni and nj are selected as the pivots for the splitting, and the clusters are split according to the statistical confidence [4].

D. Query Find Ranking Method

QueryFind, based on users' feedback and the source search engine's recommendation to provide more relevant Web pages and show them at the top of the search results list. In Query-Find, it is assumed that if a Web page is clicked many times by users and obtained high recommendation from the source search engine with a specific query word; then this Web page is relevant with this query word. First, the evaluation function of users' feedback ranking score is defined as follows:

$$F_i = \frac{C_i}{\sum_{j=1}^{n} C_j} \qquad (9)$$

In each querying relation set, Ci is the clicked times of URL i, n is the total number of different URLs, and Fi is the normalized users' feedback ranking score which is between 0 and 1. The source search engine's recommendation is used to give each Web page a content-oriented ranking score. The evaluation function of the content-oriented ranking score is defined as follows:

$$O_i = 1- \frac{(R_i - 1)}{M} \qquad (10)$$

Where O_i is the content-oriented ranking score of URL i. R_i is the original ranking order of URL i from the source search engine. M is the maximum original ranking order in this querying relation set. A Web page will have different content-oriented ranking score in different querying relation sets because the maximum original ranking order M is not the same in each set. Therefore, the content-oriented ranking score is normalized to [0, 1]

$$B_i = \frac{O_i}{\max\limits_{i=1..n} O_i} \qquad (11)$$

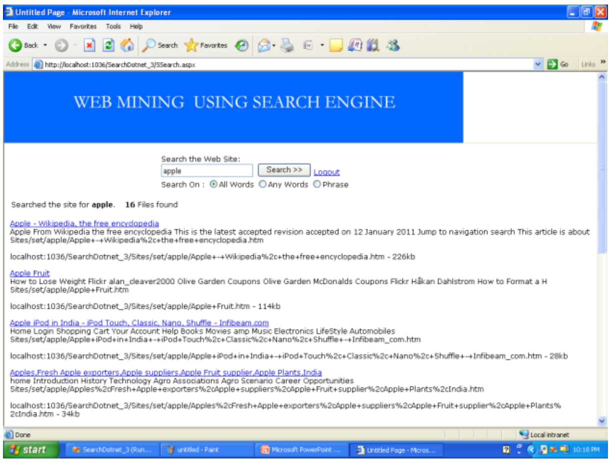
Where Bi is the normalized content-based ranking score of URL i, and n is the total number of different URLs in each querying relation set. Both the equations are combined to form the final ranking equation

$$S_i = F_i * B_i^{1/2} \qquad (12)$$

The personalized search is made effective by creating concept-based user profiles using SpyNB-C method. SpyNB-C discovers more accurate negative samples. The user profiles when integrated with Dynamic Agglomerative-Divisive clustering and QueryFind method, the results produced are more accurate to the user preference[1][6].

IV. RESULTS AND DISCUSSION

An accurate user profile can greatly improve a search engine's performance by identifying the information needs for individual users.


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:1036/SearchDotnet_3/SSearch.aspx
Go Links
start SearchDotnet_3 (Run... untitled - Paint Microsoft PowerPoint ... Untitled Page - Micros... 10:18 PM
Done
Local intranet

WEB MINING USING SEARCH ENGINE

Search the Web Site: apple
Search >> Logout
Search On : ● All Words ○ Any Words ○ Phrase

Searched the site for **apple**. **16** Files found

Apple - Wikipedia, the free encyclopedia
Apple From Wikipedia the free encyclopedia This is the latest accepted revision accepted on 12 January 2011 Jump to navigation search This article is about
Sites/set/apple/Apple+-+Wikipedia%2c+the+free+encyclopedia.htm

localhost:1036/SearchDotnet_3/Sites/set/apple/Apple+-+Wikipedia%2c+the+free+encyclopedia.htm - 226kb

Apple Fruit
How to Lose Weight Flickr alan_cleaver2000 Olive Garden Coupons Olive Garden McDonalds Coupons Flickr Håkan Dahlstrom How to Format a H
Sites/set/apple/Apple+Fruit.htm

localhost:1036/SearchDotnet_3/Sites/set/apple/Apple+Fruit.htm - 114kb

Apple iPod in India - iPod Touch, Classic, Nano, Shuffle - Infibeam.com
Home Login Shopping Cart Your Account Help Books Movies amp Music Electronics LifeStyle Automobiles
Sites/set/apple/Apple+iPod+in+India+-+iPod+Touch%2c+Classic%2c+Nano%2c+Shuffle+-+Infibeam_com.htm

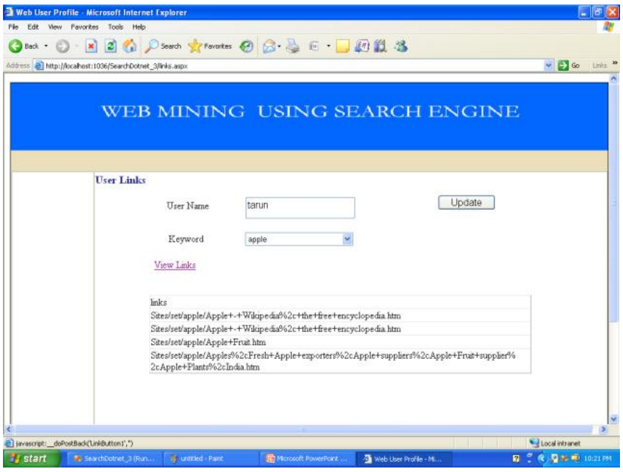
localhost:1036/SearchDotnet_3/Sites/set/apple/Apple+iPod+in+India+-+iPod+Touch%2c+Classic%2c+Nano%2c+Shuffle+-+Infibeam_com.htm - 28kb

Apples,Fresh Apple exporters,Apple suppliers,Apple Fruit supplier,Apple Plants,India
home Introduction History Technology Agro Associations Agro Scenario Career Opportunities
Sites/set/apple/Apples%2cFresh+Apple+exporters%2cApple+suppliers%2cApple+Fruit+supplier%2cApple+Plants%2cIndia.htm

localhost:1036/SearchDotnet_3/Sites/set/apple/Apples%2cFresh+Apple+exporters%2cApple+suppliers%2cApple+Fruit+supplier%2cApple+Plants%2cIndia.htm - 34kb

Figure 1: An example of clickthrough data for the query "Apple"

(This work is licensed under a Creative Commons Attribution 4.0 International License.)



http://localhost:1036/SearchDotnet_3/links.aspx

WEB MINING USING SEARCH ENGINE

User Links

User Name: tarun

Keyword: apple

View Links

links
Sites/set/apple/Apple+-+Wikipedia%2c+the+free+encyclopedia.htm
Sites/set/apple/Apple+-+Wikipedia%2c+the+free+encyclopedia.htm
Sites/set/apple/Apple+Fruit.htm
Sites/set/apple/Apples%2cFresh+Apple+exporters%2cApple+suppliers%2cApple+Fruit+supplier%2cApple+Plants%2cIndia.htm

javascript:__doPostBack('LinkButton1','')

Figure 2: An example of the user profile for the concept “Apple Fruit”.

The concept-based user profiles are created for the users by considering both positive and negative preferences. The dynamic Agglomerative-Divisive clustering (DADC) algorithm effectively exploit the relationships among the users, queries, and concepts in clickthrough data and the QueryFind ranking algorithm is used to rank the searched results

Various evaluation parameters are used to evaluate the performance of the system. The output of the system indicates that the proposed system has improved performance compared to the existing system. The Average Precision value is increased from 0.436 to 0.798 after implementing SpyNB-C. The Average Precision value of concept preference pairs obtained using SpyNB-C is 0.6925. The Average Precision value of SpyNB-C with the similar user profile sharing and ranking is 0.798.

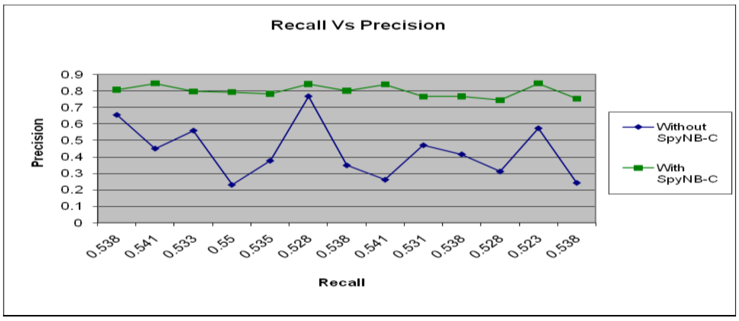


Figure 3: Precision Vs Recall graph

Future work of the system will be to use the existing user profiles to predict the intent of unseen queries, such that when a user submits a new query, personalization can benefit the unseen query. This increases the accuracy of the personalization

REFERENCES

[1] Ankur Gupta., Rajni Jindal., 2008., In Proceedings of 2nd National Development, INDIACom-2008 Computing for Feb 08-09, “An Overview of Ranking Algorithms for Search Engines”.

[2] Deng, L., Chai, X., NG,W., and Lee, D. 2004., In Proceedings of the 6th ACM SIGKDD Workshop on Web Mining and Web Usage Analysis (WebKDD 04, WA). Seattle, 71–82, “Spying out real user preferences for metasearch engine personalization”.

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

[4] Kenneth Wai-Ting Leung and Dik Lun Lee, 2010., Database Systems for Advanced Applications 5th International Conference, DASFAA 2010, Tsukuba, Japan, April 1-4, ” Dynamic Agglomerative-Divisive Clustering of Clickthrough Data for Collaborative Web Search ”.

[5] Kenneth Wai-Ting Leung, Dik Lun Lee , 2010., IEEE trans. Knowledge and Data Eng., vol. 22,no. 7,July. “Deriving Concept-based User Profiles from Search Engine Logs”.

[6] P.Wang and J.Wang, 2004., proceedings of the 2004 IEEE International Conference on e-Technology, e- Commerce and e-service (EEE’04)”Query Find:search ranking based on user’s feedback and expert’s agreement”.

[7] S. Gauch, J. Chaffee, and A. Pretschner, 2003.,ACM WIAS, vol. 1, no. 3-4, pp. 219–234. “Ontology-based personalized search and browsing,”.

[8] T. Joachims, 2002., in Proc. of ACM SIGKDD Conference, “Optimizing search engines using clickthrough data,”.

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

[10] Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. In: Proc. of ACM SIGKDD Conference (2000)

[11] Wang, X., Sun, J.T., Chen, Z., Zhai, C.: Latent semantic analysis for multiple-type interrelated data objects. In: Proc. of ACM SIGIR Conference (2006)

(This work is licensed under a Creative Commons Attribution 4.0 International License.)