A Method for User Profile Adaptation in Document Retrieval

Bernadetta Mianowska and Ngoc Thanh Nguyen

Wroclaw University of Technology,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
Bernadetta.Mianowska@pwr.wroc.pl
Ngoc-Thanh.Nguyen@pwr.edu.pl

Abstract. On the Internet the number of Web pages and other documents has grown so fast that it is very hard to find needed information. Search engines are still improving their retrieval methods but still many irrelevant documents are presented in the results. A solution to this problem is to get to know the user, his interests, preferences and habits and use this information in retrieval process. In this paper a user profile and its adaptation method is proposed. To evaluate the proposed method, simulation of user behaviour is described. Performed experimental evaluation shows that the distance between created user profile and user preferences is decreasing with subsequent actualization processes steps.

Keywords: user profile adaptation, simulation of user behaviour, document retrieval, personalization.

1 Introduction

Nowadays, with rapid growth of the Internet, large amount of information is available. Each user can use this data but the problem arises because of its overload. There is so much data and information in the Internet, that normal user is not able to find needed information in a reasonable time. Many search engines are still improving their methods of information retrieval but the user would like to get only relevant information in short time. The problem may come not only from the search engine deficiencies (such information may not exist) but also from user's lack of knowledge or skills. User may not know how to formulate a query, i.e. when he does not know about some aspects of problem or he is a new to the system [1].

Usually user enters very few words to the search engine and expects that the system will know real users' needs. To meet such expectations many systems are trying to get more information about the user and use that information during the searching process. The most popular way is to ask the user about his interests, preferences, hobbies, etc. and save them in the user profile. More complex methods are based on observation of user's activity in the system and adapt according to them. Information gathered in user profile can be used to establish real context of user's query or disambiguate the sense of it.

N.T. Nguyen, C.-G. Kim, and A. Janiak (Eds.): ACIIDS 2011, LNAI 6592, pp. 181–192, 2011.

Either explicit or implicit methods have advantages and disadvantages. Explicit methods require user's time to fill some questionnaire or look through a large set of documents to explore user's interests' areas. On the other hand, implicit method does not engage the user but need more time to decide whether he is interested in some field or not [14]. In both situations it is possible that system would not gather information about real users' interests and preferences.

The most important problem in user profiling is the fact that users' interests, preferences and habits are changing with time. To keep user profile up-to-date it is necessary to use adaptation method based on user feedback [7], e.g. documents that user opens, saves or print. The user judges relevance of documents and it has strong influence to profile adaptation.

In this paper we propose a user profile and describe an adaptation method. Proposed solutions are evaluated using simulated user. The main advantage of not using real people to the experiment is to save time but on the other hand, it requires many intuitive assumptions about user's behaviour in searching situation.

This paper is organized as follows: Section 2 contains an overview of personalization methods, problems with keeping profile up-to-date and adaptation techniques. Section 3 proposes user profile and adaptation method. In Section 4 we describe how user behaviour in searching process is simulated. Section 5 presents our experiment and discussion about the first results. Conclusions and future works are contained in Section 6.

2 Related Works

Building the user profile in personalization systems is not a new idea and many applications in this area are still developed to recommend better documents to the user. Many search engines such as Yahoo, Google, MSN, and AltaVista, are developed to meet users' needs, but they do not satisfy the various users' needs in real world. [6]. The main problem is that getting to know the user: his interests, preferences, habits, etc. and saving this knowledge in a user profile is not sufficient. Information about the user can become out-of-date because he has changed his preferences or is getting to know a new discipline.

Despite efficient information retrieval technologies, users are still not satisfied with the search process and the results presented. Studies have shown that when user is not familiar with the topic, he enters very few query terms. As a result, user is often inundated with a large amount of links returned due to the generality of the terms used during a search. It can be alleviated by providing more user information when the search was performed. [4]. System with user profile can help user in searching process by extending user's query or by disambiguating the context of it.

Approaches to user profiling can be divided into two groups according to the way of profile generation: static and dynamic. Static profile approach means that users' preferences, interests or other values are static after created the user profile. Most portal systems use the static profile approach to provide personalized information. In that case user profile can have incorrect information about the user because users' preferences have changed. To address this problem, dynamic profiles are created and various learning techniques, such as Bayesian classifiers, neural networks, and genetic algorithms have been utilized for revising user profiles in several research studies resulting in various levels of improvement [6]. A cognitive user model presented in [11] is used to characterize optimal behavioral strategies for information search using a search-engine.

In many systems, authors assume the existence of long-term and short-term interests [10]. The first category of terms are more constant and theirs changes are rarely. Short-term can change very quickly and usually are not so important for the user. In this context, the aim of personalization process is to differentiate those terms and save information that is still relevant to users' real information needs and change the influence of this information during searching process.

To build useful user profiles, many acquisition methods are generated. User enters a query and looks through the results. The system can ask the user how much each document was relevant to the query (explicit method) or observe user actions in the system (implicit method). Authors of [13] list a number of indicator types that can show if user is interested in document or not. There are following indicators: explicit – user selects from scale, marking – bookmark, save, print; manipulation – cut/paste, scroll, search; navigation – follow link, read page; external – eye movement, heart rate; repetition – repeated visits; negative – not following a link. All those features, except the first one, can be used in an implicit manner.

Different systems are generating different user profile structures and it has direct influence on adaptation and modification methods. The user profile can save information about the user in the following forms: list of historical activity, vector space model (Boolean model or with weighted terms), weighted association network, user-document matrix, hierarchical structures (tree) or ontologies [9]. Actualization of user profile depends on the profile structure. In most systems user interests, preferences, etc. are saved as set of terms with appropriate weights and terms can be linked by some relationships. Adaptation method means changing values of those weights by, sometimes complicated, mathematical formula, e.g. statistical regression or Bayesian model [14]. The way of weights modification is as important as the data, based on which, modification is done.

Important issue in profile adaptation process is not only to add new information about user but also to judge if data existing in profile is valid. The first systems like Sift NetNews obligated users to manually modify their profile by adding or removing some interests [10]. Newer solutions are based on relevance feedback and user observation. When user is not interested in some area for a long time, this area has lower influence with time and is getting unimportant in user profile. Biologically, the reasons for the forgetting model are described in [4]: amount of unrepeated information remembered by a person is exponentially decreasing with time.

User profile modification is a difficult task because information gathered about the user can not be reliable in the sense of real user information needs. That is the reason why not only positive feedback is taken into account but also the negative one [12]. The user has selected only a few documents from usually large list, and the other documents are omitted because of many possible reasons: user has found needed information in previous document, user is not interested in the other documents because they are irrelevant, user has not found it and is trying to reformulate a query or simply user does not have time to open the remaining documents.

3 User Profile and Adaptation Method

In this section we present our model for the user profile and a method of the adaptation.

Agent-based Personal Assistant is a system consisting of three modules: Personalization, Metasearch and Recommendation. The Personalization component gathers information about users' activity, interests and relevant documents. Meta Search part is responsible for finding the best search engines and sending there queries, collecting answers from all the used sources and transferring them to Recommendation part. The task of the Recommendation component is used to select and sort retrieved documents and to present them to the user [8], [9].

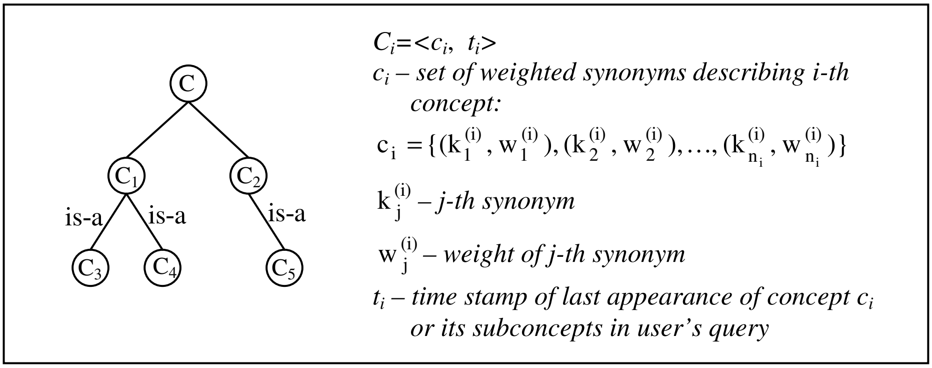
The main aim of Personalization module is to get to know the user and to build an appropriate user profile based on collected data. To guarantee that the user profile can be effectively used in the retrieval process, it should stay up-to-date. Personalization module is observing users' activities in the system and modifying the user profile.

As presented in our previous work [9] the user profile can be presented as a tree structure – acyclic and coherent graph $G = (V, E)$, where V is a set of nodes containing the vector of weighted terms describing concepts and a time stamps when the user was interested in those concepts for the last time. The E is a set of edges representing "is-a" relation between two nodes taken from WordNet ontology [17]. If two nodes v_1 and v_2 are connected by edges $e(v_1,v_2)$, it means that concept from node v_2 is a kind of concept in node v_1. Terms contained in the same concepts are synonyms. Figure 1 presents an example of user profile structure.

The root of this tree is not a concept in WordNet meaning but only an artificial node. The nodes of the first level (roots' children) are the main areas of user interests or concepts connected with them. The profile of the user with many different interests will be broad (many children of the root) and the deeper the user profile is, the more specialized information about the user the system have.

The terms in the user profile are obtained from users' queries. The intuition in this situation is that if the user is asking about some information, he is interested in it and even if there are other terms connected with users' ones, it is not obvious that the user is interested also in those additional terms.

To perform dynamic adaptation process the system observes the user and saves his queries and documents that were received as results in each session. Session is a set of users' action in the system from the opening of the system to its closure. After the



C_i=<c_i, t_i>

c_i – set of weighted synonyms describing i-th concept:

$$c_i = \{(k_1^{(i)}, w_1^{(i)}), (k_2^{(i)}, w_2^{(i)}), \ldots, (k_{n_i}^{(i)}, w_{n_i}^{(i)})\}$$

$k_j^{(i)}$ – *j-th synonym*

$w_j^{(i)}$ – *weight of j-th synonym*

t_i – time stamp of last appearance of concept c_i or its subconcepts in user's query

Fig. 1. A structure of user profile

session, from the set of documents that user has received and considered as relevant to appropriated query, the mean value of each terms from users' queries is calculated:

- Documents set from session with appropriate query:

$$D(s) = \{(q_i, d_{i_j}^{(s)})\}\,. \tag{1}$$

where: s is session's number, i is query's number and i_j is a number of documents relevant to query q_i;

- Weight of k_l-th term after s session:

$$w_d^{(s)}(k_l) = \frac{1}{n_s}\sum_{i=1}^{n_s} w_{d_i}^{(s)}(k_l)\,. \tag{2}$$

where: $w_d^{(s)}(k_l)$ is average value of weights in documents set and $w_{d_i}^{(s)}(k_l)$ is weight of term in single document in current session.

From each session system obtains a set of terms that were used in queries in this session. To calculate the weights of those terms in relevant documents, some measure of term importance in the document should be used. The most popular method to extract index terms from the document is statistical term weighting measure frequency-inverse document frequency (TF-IDF). In research collections, documents are often (but not always) described by some keywords added by the authors. Using existing keywords is more adequate than using terms artificially extracted from the document because the authors know the best way to describe their work. Some danger can be hidden in that approach when authors add words or terms that are not present in dictionary or ontology used by the system (e.g. added word is a proper name or too specialized in a very narrow research area).

After the session, new set of weighted terms are calculated. To update the weights of existing terms in profile, the change of user interests can be calculated as:

- Relative change of user interests in term k_l after the session s:

$$\Delta w_{k_l}(s) = \begin{cases} \dfrac{w_{d_i}^{(s)}(k_l) - w_{d_i}^{(s-1)}(k_l)}{w_{d_i}^{(s-1)}(k_l)}, & \mathit{if}\ w_{d_i}^{(s-1)}(k_l) > 0 \\ w_{d_i}^{(s)}(k_l), \mathit{otherwise} & \end{cases} . \quad (3)$$

where: $w_{d_i}^{(s)}(k_l)$ is a weight of term k_l after current session and $w_{d_i}^{(s-1)}(k_l)$ is a weight of term k_l after previous session.

User adaptation process is a function of two arguments: current weight of term in user profile $w_{k_l}(s)$ and relative change of user interests in two last sessions $\Delta w_{k_l}(s)$:

$$w_{k_l}(s+1) = f(w_{k_l}(s), \Delta w_{k_l}(s)) . \quad (4)$$

The following formula is proposed to calculate the weight of term k_l after the session s:

$$w_{k_l}(s+1) = \alpha \cdot w_{k_l}(s) + (1-\alpha) \cdot \left(\frac{A}{1 + B \cdot exp(-\Delta w_{k_l}(s) + C)} \right) . \quad (5)$$

where: $w_{k_l}(s+1)$ is a weight of term k_l in user profile in session $s+1$; A, B, C and α are parameters that should be attuned in experimental evaluation.

When the user is not more interested in some terms, theirs weights are decreasing and if this weight is smaller than some assumed threshold, such unimportant term is deleted from user profile.

In the next sections, we show that proposed adaptation method is effective in terms of the Euclidean measure. After the 3rd, 6th and 9th block of sessions, user profile will be updated and the distance between created user profile and user preferences will be calculated in those steps. The distance in subsequent stages will decrease so the created user profile will become closer to user preferences.

4 User Simulation

To perform an experiment instead of using a real user we would like to simulate user behaviour. We assume that users' preferences are described by set of 10 terms randomly selected from a thesaurus T. To each term a normalized weight w_i is generated $w_i \in (0{,}5; 1]$, $i \in \{1,2,...,10\}$.

Clarkea et al. [3] checked that in many retrieval systems more than 85% of the queries consisted of three terms or less. In our experiment user queries will be generated by selecting 3 terms from user preferences and will be sent to the information retrieval system. It is possible that system will not return any document to entered query because selected terms can be inconsistent [5] or simply such document can not exist in database.

User preferences can change with time so after each 10 sessions one term with the lowest weight will be deleted and another term from thesaurus T will be chosen and added to user preferences with random weight. The weights of other terms in users' preferences set will be changed by at most 10% [2], but still they will fulfill the condition $w_i \in (0{,}5;1],\ i \in \{1,2,...,10\}$. User preferences can be presented in the following form (with weights in increasing order):

$$\begin{gathered} Pref = ((t_1,w_1),(t_2,w_2),\ ...,(t_{10},w_{10})) \\ 0{,}5 \le w_1 \le w_2 \le ... \le w_{10} \le 1 \end{gathered} . \tag{6}$$

The most important feature in real user behaviour is to judge whether retrieved document is relevant to entered query or not. To do it automatically, additional weights are needed to each term of user query q and document keywords d. The best indexes are added to the document by its authors: $d = (i_1, i_2, ..., i_m);$ where m depends on how many indexes author(s) entered.

In our experiment relevance function will be used. Relevance function is calculated as a sum of differences between positions of term in the user query and position of the same term or its synonym in document indexes divided by maximal sum of those differences. The numerator of this value of proposed relevance measure can be treated as the number of operations needed to set those terms in the same order as they are in user query and on the beginning of indexes list:

$$dist(q,d) = \frac{\sum_i |pos(term_i^q)\text{-}pos(term_i^d)|}{\sum_j |num\_of\_pos(j)\text{-}1|}. \tag{7}$$

where: $pos(term_i^q)$ is a position of i-th term in user query; $pos(term_i^d)$ is a position of *i-th* term in documents' indexes and $num\_of\_pos(j)$ is a number of indexes in *j-th* document.

5 Experimental Evaluation

In this section, we discuss our methodology to perform the experiments to evaluate the effectiveness of personalization method. The aim of the experiment is to show that the distance between user preferences and generated profile is getting smaller when actualization method is applied.

The experiment was performed according the following plan:

1. Determine a domain and set of terms T that will be used in experiment. Determine user preferences *Pref.*

2. Generate a query q – select randomly 3 terms from user preferences set $\{t_1, t_2, ..., t_{10}\}$, set these terms in proper order according to decreasing weights and send query to search engine.
3. From the return list get keywords $d = (i_1, i_2, ..., i_m)$; m is a number of indexes.
4. Calculate distances *dist*(*q,d*) between each retrieved document and user query using formula (7).
5. Mark documents that meet the condition $dist(q,d) \geq \rho;\ \rho \in (0;1)$ as 'relevant'; $D = \{d_1^{(s)}, d_2^{(s)}, ..., d_{n_s}^{(s)}\}$
6. After 3 blocks of 10 session update the profile using adaptation method presented in equation (5).
7. Calculate the distance between user profile and user preferences using Euclidean distance between those two extended vectors.

Experiment was performed on ACM Digital Library [15].

The following assumptions were made for the experiment: we assume the values of weights of documents' indexes are known. The first keyword is the most important and has weight $w(i_1)=1$, weight of the next keyword is $w(i_2)=0,9$ and so on. The sixth and next keywords will have weights $w(i_m)=0,5$:

$$\begin{aligned} w(i_1) &= 1 \\ w(i_2) &= 0{,}9 \\ w(i_3) &= 0{,}8 \\ &\vdots \\ w(i_m) &= 0{,}5 \ \textit{for}\ m \geq 6 \end{aligned} \quad . \tag{8}$$

The terms from user query do not need weights. User query contains three terms and we assume that each of them is equally important. The order of those terms in query is taken into account by used search engine.

All methods described in previous sections were implemented in Java environment (J2SE). The set of terms *T* was obtained from the ACM Computing Classification System [16] and to user preferences *Pref* the following terms were selected with randomly weights:

Pref(*0*) = {(*database, 0.81*), (*network, 0.65*), (*knowledge, 0.63*), (*security, 0.9*), (*management, 0.86*), (*learning, 0.83*), (*hybrid, 0.86*), (*theory, 0.71*), (*simulation, 0.5*), (*semantic, 0.53*)}

From the set *Pref*(*0*) users' queries were generated by random selection of three out of 10 terms. In each session we assume 5 users' queries. After each block of 10 sessions, the term with the lowest weight was replaced by new term from *T* and new weight for its term was generated. The sample set of users' queries with the results (indexes of retrieved documents) obtained from the ACM Portal from one session is presented in Table 1. Each document was checked by the relevance function and from the set of relevant documents after 3 blocks of sessions, the first user profile was generated according to procedure described in Section 3.

After the very first trials, the parameters were tuned and set as follows: ρ =*0.3*; *A=1.0*, *B=1.0* and *C=3.0*. To the experiment 9 blocks of sessions was prepared. After first 3

Table 1. A sample of user session: queries and results

User query	Indexes of documents obtained for query
theory knowledge learning	knowledge management, media theory, multimedia process model, technology enhanced learning
	knowledge, knowledge management, learning organisation, management, organisational learning, theory
	collaboration, corporate universities, customer relations management (CRM), disciplinarity, educational theory, knowledge management, multimedia, organizational learning, research, training, usability studies, user-centered design
	database courseware, knowledge and skills, multimedia, tool-mediated independent learning, virtual apprenticeship theory
knowledge network security	OSPF attacks, event correlation, knowledge-based IDS, link-state routing protocol security, real-time misuse intrusion detection, real-time network protocol analysis, timed finite state machine
	computer-network security, knowledge acquisition, knowledge-based temporal abstraction, malicious software, temporal patterns
	AI reasoning, Vijjana model, cloud computing, interactive knowledge network, security, semantic web
	base stations, entity authentication, guillou-quisquater protocol, security protocols, sensor and ad hoc networks, wireless security, zero-knowledge protocol
	IHMC, MAST, concept maps, knowledge models, mobile agents, network security
learning database theory	learning theory, non-interactive database privacy
	database courseware, knowledge and skills, multimedia, tool-mediated independent learning, virtual apprenticeship theory
hybrid simulation management	failure management, hybrid embedded software, model-in-the-loop, quality assurance, simulation, simulink, testing
database management learning	LMS, Oracle, PHP, authentication, automated, database, email, instructor, lab, learning management system, online, registration, reservation, roster, training, workshop
	Case-based learning, UML, conceptual modeling, database management systems, design, education, informatics, information retrieval, lucene, postgresql

Table 2. User profile generated and updated in subsequence of 3 blocks

User profile after 3rd block		**User profile after 6th block**		**User profile after 9th block**	
term	*weight*	*term*	*weight*	*term*	*weight*
network	0,13387	network	0,28212	network	0,00000
simulation	0,14519	simulation	0,00000	simulation	0,00000
theory	0,13741	theory	0,28860	theory	0,00000
hybrid	0,12670	hybrid	0,27633	hybrid	0,41713
semantic	0,14311	semantic	0,00000	semantic	0,00000
knowledge	0,13035	knowledge	0,28021	knowledge	0,41639
learning	0,12823	learning	0,27317	learning	0,41853
security	0,13483	security	0,27616	security	0,41331
database	0,12458	database	0,26940	database	0,00000
management	0,13163	management	0,28059	management	0,41704
dynamic	0,12401	dynamic	0,27281	dynamic	0,41474
collaborative	0,11686	collaborative	0,00000	collaborative	0,00000
		storage	0,18713	storage	0,34458
		intelligence	0,18899	intelligence	0,34744
		distributed	0,18845	distributed	0,34557
				agent	0,18808
				cognitive	0,18727
				modeling	0,18675

blocks the user profile was generated and after 6-th and 9-th block, this profile was updated. Table 2 presents the user profile obtained after each update in the vector form.

To compare created and updated user profile with user preferences, the Euclidean measure was used. In the subsequent blocks of session, new terms can be added to the user profile so we need to modify this measure. To compare two vectors with different numbers of dimentions, Euclidean measure was normalized by dividing calculated value by the maximal length of vector with normalized coordinates. The results are presented in Fig.2. The distance between user preferences and created user profile is decreasing with subsequent updates.

Trends in Euclidead measure show that adapting user profile using proposed algorithm is effective and created and updated profile is becoming more similar to preferences vector, in spite of the fact that preferences are slightly changed with time.

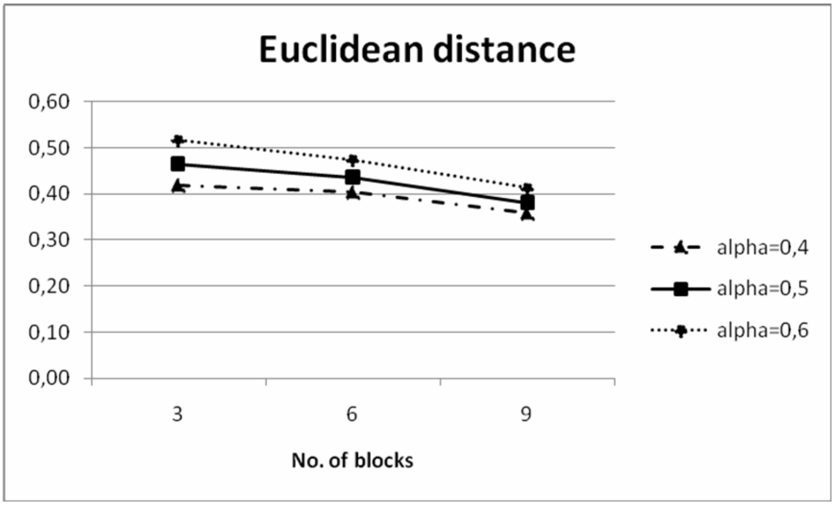


Fig. 2. Euclidean distance between user preferences and the created user profile

6 Conclusions

In this paper we present a proposition of user profile and method of its adaptation. To avoid time and work-consuming experiments with real users, the way of user simulation was presented. Performed experiment shows that updating the profile according to proposed formula is an effective way of user profiling and the adapted profile becomes more similar to user preferences.

Acknowledgments. This research was partially supported by European Union within European Social Fund under the fellowship no. MK/SN/160/III/2010/U and by Polish Ministry of Science and Higher Education under grant no. N N519 407437.

References

1. Aldous, K.J.: A System for the Automatic Retrieval of Information from a Specialist Database. Information Processing & Management 32(2), 139–154 (1996)
2. Carreira, R., Crato, J.M., Gonçalves, D., Jorge, J.A.: Evaluating Adaptive User Profiles for News Classification. In: IUI 2004, Portugal (2004)
3. Clarkea, C.L.A., Cormackb, G., Tudhope, E.A.: Relevance ranking for one to three term queries. Information Processing & Management 36, 291–311 (2000)
4. Dingming, W., Dongyan, Z., Xue, Z.: An Adaptive User Profile Based on Memory Model. In: The Ninth International Conference on Web-Age Information Management. IEEE, Los Alamitos (2008)
5. Iivonen, M.: Consistency in the Selection of Search Concepts and Search Terms. Information Processing & Management 31(2), 173–190 (1995)

6. Jeon, H., Kim, T., Choi, J.: Adaptive User Profiling for Personalized Information Retrieval. In: Third 2008 International Conference on Convergence and Hybrid Information Technology. IEEE, Los Alamitos (2008)
7. Kobsa, A.: User Modeling and User-Adapted Interaction. In: Conference Coempanion CID 1994 (1994)
8. Maleszka, M., Mianowska, B., Nguyen, N.T.: Agent Technology for Information Retrieval in Internet. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2009. LNCS, vol. 5559, pp. 151–162. Springer, Heidelberg (2009)
9. Mianowska, B., Nguyen, N.T.: A Framework of an Agent-Based Personal Assistant for Internet Users. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010. LNCS (LNAI), vol. 6070, pp. 163–172. Springer, Heidelberg (2010)
10. Mitra, M., Chaudhuri, B.B.: Information Retrieval from Documents: A Survey. Information Retrieval 2, 141–163 (2000)
11. O'Brien, M., Keane, M.T.: Modeling User Behavior Using a Search-Engine. In: IUI 2007, USA (2007)
12. Pan, J., Zhang, B., Wang, S., Wu, G., Wei, D.: Ontology Based User Profiling in Personalized Information Service Agent. In: 17th International Conference on Computer and Information Technology (2007)
13. Shen, X., Tan, B., Zhai, C.X.: Implicit User Modeling for Personalized Search. In: CIKM 2005, Germany (2005)
14. Story, R.E.: An Explanation of the Effectiveness of Latent Semantic Indexing by Means of a Bayesian Regression Model. Information Processing & Management 32(3), 329–344 (1996)
15. ACM Digital Library, `http://portal.acm.org/dl.cfm`
16. ACM Classification, `http://www.acm.org/about/class/ccs98`
17. WordNet Ontology, `http://wordnet.princeton.edu/`