2008 Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008)

Personalized Information Retrieval in Digital Ecosystems

Dengya Zhu and Heinz Dreher

Curtin University of Technology, GPO Box U1987, 6845 Perth, Western Australia,
e-mail: dengya.zhu@postgrad.curtin.edu.au, h.dreher@curtin.edu.au

***Abstract*—Search results personalization is considered a promising approach to boost the quality of text retrieval. In this paper, a personalized information retrieval paradigm is proposed which not only implicitly creates user profile by learning users' search history, search preferences, and desktop information by kNN algorithm; but also intends to deal with the problem of search concepts drift through adjusting the weight of category which represents users' search preference. By comparing the cosine similarities between vectors represent personal valued search concepts in user profiles, and vectors represent search concepts in the retrieved search results, the search results will be tailed to better match users' information needs.**

***Index Terms*—information retrieval, personalization, user profile, machine learning, kNN.**

I. Introduction

A Digital Ecosystem (DES), according to [1], is "an open, loosely coupled, domain clustered, demand-driven, self-organising agent environment, where each agent of each species is proactive and responsive regarding its own benefit/profit but is also responsible to its system." A DES is an open community, it collects information from other communities; and it also produces information consumable for other communities [1]. Therefore, effective information retrieval is essential in a sustainable DES.

However, due to the exponential growth of information in the world, and the polysemy and synonymy characteristics of natural language, effective information retrieval is in fact a nontrivial issue, especially for Web information retrieval.

Search engines are now a dominant search tool for Web information retrieval. As pointed out by [34][35], this popular search tool is now facing several challenges as discussed below. The first problem of search engines is information overload: when a user submits a short query which is a preferred format of most Web seekers, search engines usually return millions of search results. The huge number of information items is obviously information overload, because of which valuable information may be overlooked. Information overlook incurs opportunity costs.

The second problem of search engines is mismatch of search results: among the long list of Web search results, more than half of them are irrelevant [7][8][21]. When an environmentalist uses "jaguar" as a search term to search some information about animal jaguar, most of the results are about jaguar car. When a student uses "matrix" as a search term to search some information about the mathematics concept, most of the results are about the movie "The Matrix Trilogy", hairstyle, and other irrelevant items.

Missing relevant documents is another issue of search engines. This is mainly caused by the synonym aspect of natural language [6], and semantic similarity is not considered [14].

The fourth challenge stems from some clustering engines which reorganize Web search results by grouping them into hierarchical clusters. Automatically formed knowledge hierarchy can hardly match the human edited knowledge structure, such as Yahoo! Web directory [31] or the Open Directory Project [28].

The fifth issue of search engines is that search results are poorly organized in a plain-list format. This format is suitable only when the size of results set is less than 50, and the relevant documents reviewed per session are around ten [5]. Very few users will go beyond 10 pages to pick up relevant information from this long list results [12].

Another issue of information retrieval in DES is search tools and search results are not integrated into all-in-one browser [35]. In a DES, users require access to diverse information sources: the Web; personal computer; the Intranet; commercial databases; and so on. However, they usually need to use different search tools to access information from these different sources; this involves considerable trial and error and investment in mastering these different search tools.

The last issue is search results personalization, which is to be addressed in this research. As pointed out by [8], the key issue for an information retrieval system is not only to find relevant results determined by the literal similarities between queries and retrieved documents; but the information consistent with users' information needs. The value of a document can only be determined by a specific user's information needs but not other criteria.

To approach the search results personalization problem in DES, a personalized information retrieval model in DES is proposed. The model not only considers creating user profiles by learning user search history and desktop information; but also intends to leverage the power of domain oriented ontology to match user search concepts, and re-rank the search results based on the learned user profiles.

The paper is organized as follows: in section II, some work related to Web search results personalization is reviewed; section III deals with creating user profile by learning users' search preference by k-Nearest Neighbour [26] algorithm based on the ODP; section IV explains how to utilize user profiles to personalize search results; and section V concludes this paper.

II. Related work

Personalization, according to [21] concerns not only retrieving syntactically relevant information, but also a user's information consumption pattern, searching strategies, application used and the nature of the information. In this section, some previous work on personalization is re-

viewed.

[21] classifies Web search results according to three usage-granularities – individual, group/social, and general. Relevance of information [18][19] items is judged based upon their usage. Query augmentation and result processing are the two primary ways to personalize search results, where personalization elements include user's goals, prior and tacit knowledge, and past information-seeking behaviours.

User models are calculated based on the top 1000 categories of the ODP. When favourite links are imported, the corresponding pages will be fetched and classified into the ODP categories, user's weighting on the category is adjusted accordingly. If no links are imported, when a Web page is clicked, the weight in the user profile is updated the same way.

The formed user profiles are employed to augment submitted queries by comparing the search terms with user profiles utilizing the Vector Space Model (VSM) [24]. The user profiles are then used to re-rank Web search results by evaluating similarities between user profiles and the titles and other metadata from the returned pages. Experimental results show that combination of query augmentation and result processing is quite effective and efficient. However, the query augmentation in [21] needs to be further evaluated.

[8] employes Multi-Attribute Utility Theory, which is also utilized by [25], to represent user preferences as an additive value function over available metadata. Some Page-specific attributes (such as number of words per page and number of sections on a page) and the corresponding weights are manually designed. Before being submitted to meta-search engines, queries are augmented with the phrase "what is" as a prepend, and phrase "links resources" as an appendix (such as, what is Linux, and Linux links pages). The returned search results are then re-ranked based on the explicitly formed user profile. However, how to decide the dimensions / attributes, and how to automatically weight these attributes are not trivial problems.

[9] indicates the critical component in personalization is how to acquire and model user interest categories. Compared with a plain list of user interests, a hierarchical perspective of user interests perhaps better represents the human conception of a set of interests. In order to create user profiles, an incremental, unsupervised concept learning algorithm named Web Document Conceptual Clustering (WebDCC) is proposed. WebDCC aims to create a conceptual hierarchy which is a classification tree in which instances and concepts are represented by leaf nodes and internal nodes respectively. Root node represents the most general concept; the offspring node represents more specific concept which summarizes all instances classified under it. Web documents are represented by normalized tf[1] scheme, whereas idf [23] factor is not considered because of the incremental characteristic of WebDCC. The similarities between a given instance and the formed concepts are measured by the vector space model [2][24]; and kNN algorithm [3][32] is employed to determine which instance clusters the given document should belong to. Experimental results demonstrate that the performance of WebDCC is comparable to that of agglomerative hierarchical clustering algorithm. However, the incremental feature of WebDCC makes it inherently affected by the order in which instances are presented.

[4] explores the ODP metadata for Web search personalization. The ODP data are organized in a tree-like data structure, and by using reference (symbolic) links, some nodes can appear have more than one parent[2]. In the ODP, categories are organized as internal nodes and Web pages are always in leaf nodes. User profiles are created simply by asking users to select a list of categories which best match their search interests. When a list of search results is returned from a search engine, the similarities between each result and the categories in the user profile are compared, and the search results are re-ranked based on the new calculated similarities. The drawback of [4] is that user profile is very simple and the users have to be interrupted to select from a long list of categories which fit their search interests.

[7] indicates user profile should not be explicitly created for the reasons that firstly, it imposes an extra burden on users; secondly, the interest description of users may not be accurate; and thirdly, users' interests change over time whereas few users update user profile correspondingly. User profile may be created by "watching over users' shoulder" [7]. They indicate that user profile is essentially a reference ontology in which concepts are weighted based on a user's perceived interest in terms of Web page content, length of the page and time spent on the Web pages.

Hierarchy of a Web directory, such as the ODP, Yahoo! Web directory, or Lycos [16], can be used as a reference ontology to represent user's search interests. Only the concepts of the top four levels of the subject hierarchy which consists of 4,417 concepts are used to express users' search interests, because this large number of concepts is adequate as training data.

VSM are utilized by [7] to classify Web pages visited by users into concepts in the reference ontology. The contents of Web pages manually linked to each concept in the subject hierarchy are concatenated to form a super-document (D) which is then pre-processed by removing stopwords and by stemming. The cosine similarities (Sc) between vectors represent these super-documents and the Web pages from users' Web browser cache folder are compared. The concept (represented by D) with the highest similarity value is supposed as most relevant to the Web page, and the similarity values (Sc) are accumulated to the top five concept's weights. Considering time spent and document length factor, TL = time/length, or log(time/log(length)), the Sc is revised as S = Sc*TL. Experimental results of [7] demonstrate that the number of a user's concepts of interest converges over time, and after the original search results are re-ranked and filtered based on the formed user profile, an eight percent performance increase is achieved.

[1] tf-idf: term frequency/inverse document frequency. This is a term weighting scheme based on statistical information regarding occurrence of indexed terms in the document space [2].

[2] A node in a tree has at most one parent

2008 Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008)

The effectiveness of classification algorithm selected is one drawback of [7]. Another issue is the chosen of some numbers, such as the top **five** concept's weights, is need to be further supported by extensive experimental data.

Time spent on a Web page is also an important factor when modelling a user profile. [27] believe requested pages, content-related meta-data, user session information, and structure information are the main factors to create a user profile. In addition, the time a user spent on a specific page, the frequency of how often the user requests a specific page, the centrality (a page has short paths to all other nodes) of a page, and the prestige of a page (how often a page is referenced by other pages) are all factors to be considered. However, [27] argue that the time a user spent on a page is more important than the other factors, and they weight time, frequency, centrality and prestige to 70%, 20%, 5% and 5% respectively. Data in log file are extracted to form a graph and a corresponding adjacency matrix which indicates the click paths of a user. Pages required are then analyzed to extract meta-tags to form a knowledge structure; important pages are used to tailor the formed knowledge structure to create a final user profile.

In Web usage mining, in addition to the factors such as the order of visited Web pages and the popularity of the pages, [11] argues that the time spent on Web pages (TSP) is also an important measure of information intention and relevance. TSP can be obtained by analyzing the server-side log file, which strongly depends on domain and pre-processing – filtering out robot transactions and session identification. However, issues like user distraction and effective reading time on a page need further study.

[35] proposes an Integrating Text Retrieval Framework (ITRF) which aims at providing effective information retrieval services for digital organisms in a DES by leveraging the power of Web searching technology. In ITRF, user profile based on IMS Learner Information Project (LIP) [15] is proposed because LIP emphasises the facilitation of information flow between computational systems.

III. User Profile Creation by Learning Search Concept

The first question to be asked before creating a user profile is which approach is appropriate. User profiles can be created explicitly [4] or implicitly [7], or by a combination of the both [21]. Search preferences and interests in explicitly created user profiles are directly indicated by users, and thus simple to create and should be accurate. However, the 'explicit' approach suffers from extra burden on the user; inaccurate interest description and the risk of becoming outdated because few users update their profiles as their interests change over time [7]. Therefore, in this research, user profile is created by machine learning techniques [30], with an optional function which enables users to override the automatically created user profile to explicitly express their search preferences and interests with a relativity indicator or scale. This is different from the approach of [21] which only tries to import favourite links stored in a search browser.

A. Factors in Personal Reference Modelling

When machine learning approach is selected to build user profile, the question followed is how to decide the attributes which depict a user profile. As can be seen from the related work discussed in the previous section, a user model can be described from different perspectives with different aspects. For example, [20] suggests the interests, the knowledge, the objectives, and the preferences of users to be the four elements to describe the user model. However, most of the machine learning related research considers only the Web usage mining which involves the content of Web pages visited, in addition to the time spent and the size of the pages [7]. Web usage mining is becoming a dominant approach for personalization [20]. In this research, in addition to Web usage mining, we offer a mechanism to utilise the information stored in personal computers for modelling users' search preferences, a feature which few researchers have taken into account.

B. Learning User Preference based on the ODP

The ODP is the largest, most comprehensive human edited Web directory. More than 57 thousand editors have created more than 590 thousand categories with nearly five million submitted Web pages[3], each of which contains the title and the brief description of the page. Categories in the ODP are organized as a tree-shaped structure with symbolic links; one node is allowed to have more than one parent [4]. A category node near the root of the tree represents a more abstract concept and its offspring category nodes represent more specific concepts.

Categories in the top two levels of the ODP are employed to represent users' preferred search concepts [34], while [7] uses categories of the top four levels. There are about six hundred ODP categories in the top two levels, and this is enough to distinguish users' search preference for personalization purposes in a two level hierarchical knowledge framework.

User profile in this research is actually a reference ontology mapped directly from the ODP, as shown in Fig. 1. Categories in the top two levels of the ODP serve as a knowledge framework in the user profile. In the very beginning, each category is weighted to zero indicates this category is not a preferred search concept. The learning process is to assign a weight to the corresponding category by exploring information stored in the user's personal computer and Web data mining. The weight of a category in the reference ontology is then used as an indicator of a user's search preference on this category. All the factors related to personalization are finally mapped and represented by the weights of the categories. Weighting a category in the reference ontology is the core of user profile creation.

Machine learning is an effective approach for user modelling. However, as pointed out by [30], the following four issues need to be considered when using machine learning algorithms for user modelling, that is**,** the need for large data sets, the need for labelled data, conceptual drift, and

[3] the figures come from the Web site of the ODP on 25 Oct 2007

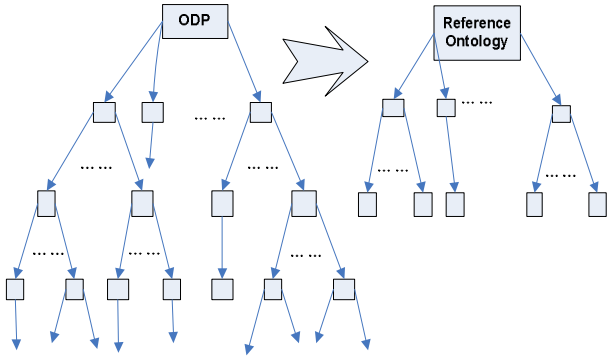

Fig. 1 the ODP structure and the reference ontology

computational complexity.

To address the first problem, information stored in personal computers is explored based on the intuition that people tend to store their preferred information in their computers. The information will be stored with some kind of extensions such as Microsoft word, PDF, or HTML and so on. However, the data under the folders such as "program files", "Microsoft" "Windows" and the like will not be considered because they usually contain common computer software which does not suggest users' search preferences.

Google Desktop SDK [10] is available to manage information in personal computer environments. It can crawl, index, cache, and search content on the personal computer. The crawled desktop information is used as a source to build user profile.

Users' Web usage information is also analyzed in the user profile learning process. Whenever a user conducts a search action, the information related to this action, including the content of the visited Web page, the time spent, the size of the Web page, are all taken into account to estimate the weight of a category in a user profile. Details will be described in the following section.

The second issue, the need for labelled data, is addressed by exploring the data in the ODP. As discussed in [34], the semantic characteristics of a category in the ODP can be manifested by a category-document composed of the topic of the category, the description of the category, and the submitted Web pages under this category where each page has a title and a brief description. Because each category-document is "assigned" to a category, this category-document set can be regarded as a labelled training data set authorized by human experts. With this large amount of training data available, the need for labelled data set is catered for.

The third issue of user profile creation is concept drift [13][29] which indicates users' search preferences and interests are not stable but change over time. Users' search interests with regard to time are assumed to follow a normal distribution where time t is positive. In this research, concept drift is dealt with by adjusting the weight of category in the user profile, as shown in Fig. 2, where the vertical axis represents the probability of search preference, and the horizontal axis represents the time variable.

Users' search activities at this research stage are simply divided into three phase: current, recent, and historical. For a search concept $\mathbf{d}_i$, suppose that the corresponding weight in the reference ontology is $\mathbf{W_j}$, and then the time-adjusted weight is

$t\_w_i = u(t)*\mathbf{W_j}$

where u(t) reflects time related user search preferences. t maybe take discrete values such as *current*, *recent*, and *historical*. To simplify the user model, let

$$u(t) = \begin{cases} 0.95 & \textit{current} - \text{most current 500 searches} \\ 0.75 & \textit{recent} - \text{past 501 – 2000 searches} \\ 0.3 & \textit{historical} - \text{searches earlier than 2000} \end{cases}$$

Note that the definition of the three phases can also be based on other criteria - searches within last three months can be regarded as current; searches earlier than three months and no earlier than one year are in recent; searches one year ago are taken as historical.

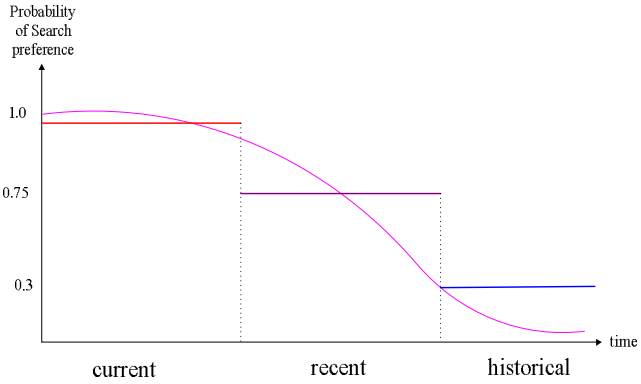

Fig. 2 time factor in user modelling

It is not necessary to adjust t_w_i in synchronicity with each search activity of users' for the purpose of efficiency. For example, time-adjusted weight can be calculated every five or ten searches to reduce the time spent on computation. This elicits the last issue to be addressed.

The last issue of using machine learning for user modelling is computational complexity because many effective learning algorithms are time consuming. In this research, after considering the tradeoffs between effectiveness and computational complexity, the kNN algorithm is selected [26][33]. Details of learning process are presented in the following section.

C. Using kNN Algorithm to Learn Search Concept Preferences

To utilize kNN algorithm [17], each category-document is indexed with the tf-idf weighting schema [2][23] which aims at determining features that better describe a document in a document set, and features that better distinguish the document from the rest documents in the document set. Let N be the total number of documents in the document set, T be the total number of terms which appear in the document set, n_i be the number of documents in which the index term t_i appears, $freq_{ij}$ be the frequency of term t_i in document d_j, and $M_j(freq_{ij})$ is the maximum number of term frequency in document j. The normalized term frequency tf_{ij} is given as [2]

$$tf_{ij} = \frac{freq_{ij}}{M_j(freqi_j)}$$

the inverse document frequency for term t_i is given by

$$idf_i = \log \frac{N}{n_i}$$

the if-idf weighting scheme is presented as

$$w_{ij} = tf_{ij} \times \log \frac{N}{n_i}$$

To reduce the high dimensionality of term space, indexed terms are stemmed [22] and stop words[4] are removed. After this process, all category-documents are indexed and ready to be served as training data. All Web pages visited by users and all documents crawled in personal computer are indexed the same way.

To train the user profile with the prepared training data, a distance function defined in Euclidean space is needed. In this research, vector space model [2][24] is employed. In VSM, the similarity between two objects is measured by the cosine value of the angle θ between the two vectors. Let $\vec{c}, \vec{d}_j$ represents vectors of the Web page, or a document stored in personal computer, and a category-document; the similarity function is then defined by [24] as $\cos(\theta) = \text{sim}(\vec{\mathbf{d}}_\mathbf{j}, \vec{\mathbf{c}})$ which measures the search concept similarity between $\vec{c}$ and the category-document $\vec{\mathbf{d}}_\mathbf{j}$ representing a category in the ODP.

Using the $\text{sim}(\vec{\mathbf{d}}_\mathbf{j}, \vec{\mathbf{c}})$ result of above, in this research it is proposed to make the following adjustment. Select k maximum cosine values, and then vote the majority category and assign it to $\vec{c}$ to finish concept learning. The cosine similarity value is thus given as

$$cs_j = \frac{1}{m} \sum_{i=1}^{m} \text{sim}(\vec{\mathbf{d}}_\mathbf{j}, \vec{\mathbf{c}}) \quad \text{..... Equation (1)}$$

where *m* is the majority vote.

However, the majority sometimes does not exist for the first round selection. In this case, for the k selected categories, their parent categories will be considered, as illustrated in Fig. 3. Suppose k = 5 and the five top similar categories for a given document are nodes H, I, J, L and M; node E has only two supporters I and J, other nodes have only one supporter, no majority node is selected; consider their parents nodes, node A has three supporters and thus selected as majority.

Time spent on a Web page and the length of the page are another two factors which are considered reflecting users' search interest. Suppose *ts* is the amount of time spent on a page, and *len* is the length of the page in bytes, according to [7], the time-length factor can be measured by

$$tl = \log \frac{ts}{\log len}$$

Using log*len* can reduce the impact of length of a page on the similarity measurement; users are usually very quick to decide if a document is relevant or not, no matter how long the Web page is.

Now, three weights related to user search preferences and interests are obtained. A final weight $\mathbf{W_j}$ is thus given by

$$\mathbf{W}_j = cs_j \times tl \times u_i$$

So far, for a visited page or a document in personal computer, the preference weight $\mathbf{W_j}$ for document j has been obtained. In the user profile, the corresponding category in the reference ontology is added to $\mathbf{W_j}$. If category j in the ODP is deeper then two levels, its parent at level two in the ODP is the corresponding category in the reference ontology.

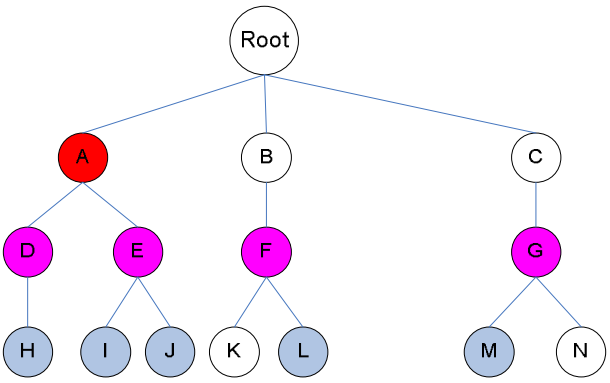

Fig. 3 majority voting

The last process is to normalize these weights. Let M be the total number of categories in the user profile, the final weight for a category in a user profile is assigned by

$$\mathbf{W}_j = \frac{\mathbf{W}_j}{\sqrt{\sum_{i=1}^{M} \mathbf{W_i^2}}} \quad \text{.....Equation (2)}$$

IV. Personalize Search Results

Personalization based on learned user profile involves a two-pronged approach: search term refine/augmentation and search results re-organization.

A. Search term refine/augmentation

To boost the recall of search results, in addition to the search-terms submitted by users, an augmented query is also submitted. The new query if formed by prepending "what is" to users' query when the users' query is only one or two words. In this case, the user is supposed to search information related to the concept represented by the query. [8] also appends a "links resources" to a user's query which is not suitable in this study.

B. Search Results Re-organization

For each returned Web results c_j, the cosine similarities cs_{ij} between c_j and category-documents d_j is estimated by using $\text{sim}(\vec{\mathbf{d}}_\mathbf{j}, \vec{\mathbf{c}})$ given by [2]. Utilizing majority voting algorithm discussed above, a category in the ODP can be obtained, and the similarity is updated to cs_j by Equation (1). Mapping this category to the corresponding category in reference ontology in user profile, the final personalized similarity is estimate by

$$ps_j = cs_j * W_j \quad \text{..... Equation (3)}$$

where cs_j is determined by Equation (1), and W_j is determined by Equation (2), represents the weight of personal

[4] a list of stop words is available to download from Wikipedia, Oct, 2007)

search preference on that category d_j. Personalized search results are re-organized by the descending order of *psj*.

C. Recommendation based on User Profile

Search results are categorized into different categories d_j by the kNN learning algorithm. Mapping these categories in a user profile, the corresponding user preference weight W_j on d_j can be obtained by Equation (3). Three categories with the highest W_j are recommended to users. When a recommended category is selected by users, only search results classified under this category are presented to the user.

V. Conclusion

In this paper, a personalized search algorithm is proposed which implicitly learns users' search preferences and interests by kNN. The proposed algorithm considers not only Web usage mining as a personalization tool, but also takes account to information stored in desktop computers. The problem of concept drift is addressed by adjusting the weight of users' search preferences in via a reference ontology in the user profile.

VI. Acknowledgement

This work is partially supported by Digital Ecosystems and Business Intelligence (DEBI) Institute of Curtin University. Thanks to Professor Elizabeth Chang for her advice and support.

VII. References

[1] H. Boley and E. Chang, "Digital Ecosystems: Principles and Semantics", in *Proceedings of the Inaugural IEEE Digital Ecosystems and Technologies Conference*, 2007, pp. 401-406.

[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, New York & Addison Wesley, Harlow, 1999.

[3] S. Chakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*, Morgan Kaufmann, San Francisco, 2003.

[4] P. A. Chirita, W. Nejdl, R. Paiu and C. Kohlschutter, "Using ODP Metadata to Personalize Search", in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005, ACM Press, New York, 178-185.

[5] G. G. Chowdhury, 2004, *Introduction to Modern Information Retrieval*, 2nd edt, Facet Publishing, London.

[6] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman, "Indexing by latent semantic indexing", *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990, pp. 391-407.

[7] S. Gauch, J. Chaffee and A. Pretschner, "Ontology-based personalized search and browsing". *Web intelligence and Agent System*, 2003, vol. 1, no. 3-4, 219-234.

[8] E. J. Glover, S. Lawrence, M. D. Gordon, W. P. Birmingham and G. L. Giles, "Improving Web searching with user preferences. Web Search – Your Way", Communications of the ACM, December 2001, vol. 44, no. 12, pp. 97-102.

[9] D. Godoy and A. Amandi, "Modeling user interests by conceptual clustering", *Information Systems*, 2006, vol. 31, 247-265.

[10] Google Desktop SDK, http://desktop.google.com/dev/indexapi.heml

[11] P. I. Hofgesang, "Relevance of time spent on Web pages", in *Proceedings of the WebKDD 2006: KDD Workshop on Web Mining and Web Usage Analysis (KDD 2006)*, August 20-23, 2006, Philadelphia, PA.

[12] B. J. Jansen and A. Spink, "How are we searching the World Wide Web? A Comparison of Nine Search Engine Transaction Logs", *Information Processing and Management*, 2006, vol. 42, pp. 248-263.

[13] I. Koychev, "Learning about User in the Presence of Hidden Context", in *Proceedings of the UM 2001 Workshop on Machine Learning for User Modeling*, pp. 49-58.

[14] Y. Li, Z. A. Bandar and D. McLean, "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources", *IEEE Transactions on Knowledge and Data Engineering*, July/August 2003, vol. 15, no. 4, pp. 871-882.

[15] LIP, IMS Learner Information Package, http://www.imsglobal.org/

[16] Lycos, http://www.locos.com

[17] T. M. Mitchell, *Machine Learning*, McGraw-Hill Companies, New York, 1997.

[18] S. Mizzaro, "Relevance: The Whole (hi)story", *Journal of the American Society for Information Science*, 1997, vol. 48, no. 9, 810-832.

[19] S. Mizzaro, "How many relevances in information retrieval", *Interacting with Computers*, June, 1998, vol. 10, no. 3, 303-320.

[20] D. Pierrakos, G. Paliouras, C. Papatheodorou, C.D. Spyropoulos, "Web Usage Mining as a Tool for Personalization: a Survey", *User Modeling and User – Adapted Interaction*, Nov. 2003, vol. 13, no. 4, 311-372.

[21] J. Pitkow, H. Schütze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar and T. Breuel, "Personalized Search: A contextual computing approach may prove a breakthrough in personalized search efficiency", *Communications of the ACM,* September 2002, vol. 45, no. 9, 50-55.

[22] M. F. Porter, "An Algorithm for Suffix Striping", *Program*, 1980, vol. 14, no. 3, pp. 130-137.

[23] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval", *Information Processing & Management*, vol. 24, no. 5, 1988, pp. 513-523.

[24] G. Salton, A. Wong and C.S. Yang, "A Vector Space Model for Automatic Indexing": *Commmunication of the ACM*, Nov. 1975, vol. 18, no. 11, 613-620.

[25] R. Schafer, "Rules for Using Multi-Attribute Utility Theory for Estimating a User's Interests", *in Proceedings of the 9th GI-Workshops: Agent Based Information Systems (ABIS)*, 2001.

[26] F. Sebastiani, "Machine Learning in Automated Text Categorization", *ACM Computing Surveys*, 2002, vol. 34, no. 1, 1-47.

[27] G. Stermsek, M. Strembeck and G. Neumann, "A User Profile Derivation Approach based on Log-File Analysis", in *Proceedings of the International Conference on Information and Knowledge Engineering (IKE'07)*, June, 2007.

[28] The Open Directory Project, http://www.dmoz.org

[29] A. Tsymbal, "The problem of concept drift: definitions and related work", Technical Report TCD-CS-2004-15, Trinity College Dublin, 2004.

[30] G. I. Webb, M. J. Pazzani and D. Billsus, "Machine Learning for User Modeling", User Modeling and User-Adapted Interaction, 2001, vol. 11, no. 1-2, pp. 19-29.

[31] Yahoo Web Directory, http://dir.yahoo.com

[32] Y. Yang, "Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval", *Information Retrieval*, 1994, vol. 1, no. 2, 69-90.

[33] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization", *Information Retrieval*, 1999, vol. 1, pp.69-90.

[34] D. Zhu, "Improving the Relevance of Search Results via Search-term Disambiguation and Ontological Filtering", 2007, Master Thesis, Curtin University.

[35] D. Zhu and H. Dreher, "An Integrating Text Retrieval Framework for Digital Ecosystems Paradigm", in *Proceedings of the Inaugural IEEE Digital Ecosystems and Technologies Conference*, pp. 367-372.