SEARCHY: An Agent to Personalize Search Results

Ivan Marcialis and Emanuela De Vita

***Abstract*—User's behaviour in browsing sessions is a valuable source of information useful to analyze user interests and personalize the human-computer interaction during information seeking tasks. In this paper we present a simple way to machine-learn user profile and employ it to improve Web searches. Searchy is a personal agent, embedded in the Web browser, able to sense user's tastes, to assist her/him during query formulation phase and, finally, to filter and sort the results according to the user's needs. Searchy uses the DART P2P network to store user's profiles in order to enable collaborative recommendations.**

***Index Terms*—Cooperative systems, feedback, information retrieval, user modeling.**

I. Introduction

THE World Wide Web is the largest and most widely known and used document repository, but in some respects it has become a victim of its own success. More than 150 million[1] Web sites are on line at the moment, and while the amount of available information is rapidly growing, users are in trouble when they have to find the one they need. Search Engines, the most popular tool for information retrieval, do not always meet users' expectations: document retrieval based only on the use keywords to perform the query, does not take into account that different users, while submitting identical search terms, seek different results. Studies have shown that users enter very few query terms[2], especially when they are not familiar with the topic.

User profiling approaches and filtering techniques are widely used to improve the precision of search facilities and to solve the problem of information overload, but their application to Web search is not widespread and is often used to build a customer profile.

Search Engines are not designed to offer results according to user's individual need: people, neither words nor algorithms, would play the central role in Web search.

Manuscript received January 24, 2008.

DART (Distributed Agent-Based Retrieval Toolkit) project has been partially funded by the Italian Ministry of Education, University and Research, contract grant number 11582.

I. Marcialis and E. De Vita are researchers with the CRS4, in the Network Distributed Applications Area. Email: {ivan, emy}@crs4.it. Address: Parco Scientifico e Tecnologico, POLARIS, Edificio 1, 09010 PULA CA - Italy. CRS4 Home Page: http://www.crs4.it. DART Project: http://www.dart-project.org.

[1] In the December 2007 survey we received responses from 155,230,051 sites. This is an increase of 5.4 million sites since last month, continuing the very strong growth seen during this year; the web has grown by nearly 50 million sites since December 2006. http://www.netcraft.com/

[2] C. Bradford and I. W. Marshall "Analysing users WWW search behaviour" *in Proc IEE colloquium* 99/149 - Navigating the web

In this paper, we present Searchy, a software agent developed to assist users during Web searches: Searchy analyzes user's behaviour in browsing session, it identifies the relevant Web resources and, through content-based algorithms, it deduces which topics are interesting for the assisted one. Searchy uses these data to build her/his User Profile (UP) and finally, shares it with the Web Community.

The UP is modeled after an ontology of topics, which is also used to describe every resource in our system: *Users, Documents* and *Query Strings* are all described as weighted vectors in a space of categories, and this allows Searchy to compare numerically users, pages and queries. Searchy exploits this knowledge to order and filter the result set according to UP and to identify other users, with similar UP, who are likely to be able to suggest interesting items.

These functionalities are provided by a module embedded in the Web browser, whose behaviour can be customized to meet user's needs.

This work is part of DART, a project focused on realizing a distributed and collaborative engine for semantics and personalized searches on the Web.

In the next section we discuss related works; in the third we give a brief overview of the project DART. Following this, we present in detail the architecture of Searchy and in the seventh section we discuss the experimental evaluations. In the last section we present how we planned to carry on this research.

II. Related Works

We based our work on different personalized systems, especially on systems developed to improve search results, such as Obiwan [21] and Persona [25], which were our main points of reference. We also studied systems to help users in browsing the Web, such as Letizia [14] and WebWatcher [4], and Usenet news, such as GroupLens [12]. We also investigated systems specialized in different domains, such as WebMate [8] and SmartPush [13] –which are designed to find news articles– and Quickstep [17], which is specialized in finding scientific and research papers. We gave great importance to Syskill&Webert [20], a software agent able to perform a combination of all these activities.

User profile can be created using different types of data. Letizia, WebMate, WebWatcher and Obiwan build the profile analyzing the content of Web pages visited by the user; Basar [26] uses the bookmarks list; Syskill&Webert and Persona, queries and search results. Dumais et al. [9] combine information from multiple sources (emails, Web pages, local stored documents, etc.).

Profiles may be represented in a variety of ways. Letizia

978-0-7695-3163-2/08 $25.00 © 2008 IEEE
DOI 10.1109/ICIW.2008.86

and Basar produce a list of URLs; Syskill&Webert and Quickstep create a simple list of concepts of interest; an ontology is built by SmartPush and Persona. Apart from the model used, the system's misunderstandings produce a certain amount of noise; the irrelevant concepts can be removed from the profile in different ways. Some systems require user feedback, for instance Syskill&Webert, Basar, SmartPush and Persona; others, such as Letizia, WebMate, WebWatcher, adapt autonomously.

Among recent studies of personalized systems research, we mainly refer to PVA system [7]. It uses browsed Web pages to create a concept-hierarchy-based user profile. The profile is learned by classifying the Web pages using the vector space model and adapted by merging/splitting concepts in the profile. G. Xu at al.'s work [28], about collaborative recommendation, was widely used in our research too.

III. SEARCHY IN THE CONTEXT OF THE DART PROJECT

DART (Distributed Agent-based Retrieval Toolkit) [1], [2] is a research project, focused on realize a distribute architecture for semantics searches on the Web and the access to personalized contents. Also the project aims at supply position based information strictly related to a user indicated area, so to provide spatial queries based on geo-referenced data.

The users who choose to join actively the community have to install on their PC the system back-end, called DART Node. These users are supposed to contribute to the system in terms of storage and CPU cycles, but also sharing information as P2P applications[3] use to share assets. "We use a novel indexing data structure called RDHT (Range capable Distributed Hash Table) derived from skip lists and specifically designed for storing and retrieving geographic data from a structured P2P network overlay." [24]

The nodes include a module responsible for all the semantic activities [3]. Thanks to a linguistic analysis and a semantic interpretation performed by means of Natural Language Processing (NLP) techniques, this module parses the given text to return the list of relevant words associated with their possible meanings and describes the text with a set of weighted categories.

Searchy is the DART front-end, in the following sections we present our user agent, its features and their implementation.

IV. BUILDING THE USER PROFILE

The UP is a weighted ontology that is automatically and implicitly filled in while the user browses. Searchy, installed as a Firefox extension, generates profiles by analyzing the user's surfing behaviour, specifically the contents viewed, the time spent and the actions performed on each Web page the user visits. The Web pages are automatically classified with respect to the concepts defined in the reference ontology. The results of the classification are accumulated in proportion to what the user implicitly pointes out to be interesting. Explicit feedbacks, even though allowed, are not necessary.

$$f(a) = \frac{V - A}{V} \quad \text{where} \quad \begin{array}{l} a = \{bm, pr, dl, sn\} \\ V \text{ is the number of the viewed pages} \\ A \text{ is the occurrence of "a"} \end{array} \tag{4}$$

A. Interesting Web sites

Identifying interesting Web sites is our agent's main goal. We chose a hybrid approach: it is mainly based on the study of user's behaviour [10], but it also allows explicit feedback to correct the relevance automatically assigned. We call User Relevance (*uR*) the value obtained by summing implicit and explicit feedbacks.

Previous research has shown that the effectiveness of some implicit measures of interest may be dependent on the task at hand. For instance, time spent reading has been identified as a good indicator of interest for news reading [12], [19] and Web browsing [6], [22], but contradictory results have been found for information retrieval tasks [11], [27].

In light of these works, we have decided to assign to the time spent reading the page only 50% of the *implicit uR*. The remaining contribution comes from the other actions that we consider (1).

$$uR_i = \frac{y_{1i} + y_{2i}}{2} + c_i \quad \text{where} \quad \begin{array}{l} 0 \le y_{1i} \le 1 \\ 0 \le y_{2i} \le 1 \\ c_i = \text{explicit feedback} \end{array} \tag{1}$$

The Time of Reading (ToR) value, in the previous function called y_{1i}, is calculated as showed in (2) where x_{1i} is the total seconds spent reading the page *i-th*, and k_1 is the total seconds spent reading the most viewed page. The ratio between x_{1i} and k_1 is raised to the second power to reward the pages where the user spent more time and, also, to bring the others near zero.

$$y_{1i} = \left(\frac{x_{1i}}{k_1} \right)^2 \tag{2}$$

The Executed Actions (ExA) value, called y_{2i}, is similarly calculated (3). x_{2i} is calculated by summing the values associated with every considered action, executed on the page *i-th*: bookmarking (*bm*), printing (*pr*), saving (*dl*) and sending (*sn*).

$$y_{2i} = \left(\frac{x_{2i}}{k_2} \right)^{\frac{1}{2}} \quad \text{where} \quad \begin{array}{l} x_{2i} = f(bm) + f(pr) + f(dl) + f(sn) \\ 0 \le x_{2i} \le 4 \\ k_2 = 4 \end{array} \tag{3}$$

We value every action as showed in (4).

[3] Azureus, Java BitTorrent Client (http://azureus.sourceforge.net/) and eDonkey2000, Overnet (http://www.edonkey2000.com/)

In this way, for example, the *printing* action value *f(pr)*, will be near to zero for the user who usually prints a lot of pages; in opposite, *f(pr)* will tend to 1 for the user who, in his DART history, has printed only one page.

The ratio between x_{2i} and k_2 is raised to one half to reward every page on which almost an action has been executed.

The *implicit uR* can be corrected by explicit feedback: banning and voting. The first one sets *uR* at zero. About the second one, Searchy allows the users to vote pages with a value from 1 to 5; the *uR* is summed by the highest User Relevance (uR_{max}) multiplied by a value associated with the vote (Table 1).

In this way, Searchy is able to assign to every textual resource that the user has reached on the Web, a relevance value which tries to establish how the resource is interesting for the user.

TABLE I
VOTING ACTIONS VALUES

Vote	Value = *c*
1	$-uR_{max} * 0.20$
2	$-uR_{max} * 0.05$
3	$uR_{max} * 0.02$
4	$uR_{max} * 0.07$
5	$uR_{max} * 0.15$

B. Reference Ontology

We chose to base our ontology on an already existing subject hierarchy: WordNet Domains [15], [16] provides a simple Web page ontology; it consists of 167 categories[4] with which the WordNet [18] synsets[5] have been manually annotated.

Every time the user browses a page, Searchy extracts the text and applies to the Dart Node for the classification, we will explain in the sixth section how Searchy communicates with the back-end. The Node refers the text analysis to the Semantic Module which returns the related set of weighted categories. The description of this module is not topic of this paper, we can just say that: it integrates the Link Grammar parser [23] to identify the syntactical structure of sentences, and to resolve the terms' roles ambiguity in natural languages; it analyzes each sentence identifying roles, meanings of terms and semantic relations in order to extract "part of speech" information, the synonymy and hypernym relations from the WordNet semantic net; and, finally, it classifies the text on the reference ontology, proposed by WordNet Domains.

Whether the DART Node has not yet parsed the page, Searchy adds the page's URL to the Shared Queue of URIs (SUQ) to be crawled. In this way we can order the SUQ on the base of Searchy requests, giving priority to the most visited pages rather than the most linked.

Moreover, because Searchy is client side and installed into the browser, it can analyze every resource the user views, including personalized and restrict access pages, that, in all likelihood, are the most useful to describe the user.

C. The user model

Searchy, after assigning the *uR* to viewed resources and receiving from the Semantic Module the related sets of categories, is able to build the user profile. We chose the same model used to describe every resource into the DART Community: an ordered list of weighted categories, based on the reference ontology.

The assistant fills the model as showed in (5) and save it using APML[6] specifications. APML stands for Attention Profiling Markup Language; it is an XML-based format for capturing a person's interests and dislikes and storing them as a list of the topics and sources the user is interested in, and a value representing his level of interest in them.

$$P(c_j) = \sum_{i=1}^{m} c_{ji} \cdot uR_i \tag{5}$$

where
$P(c_j)$ = *weight of j-th category c in profile P*
c_{ji} = *weight of j-th category c for i-th resource*
uR_i = *user Relevance of i-th resource*
m = *number of viewed resources*

Searchy stores in the RDHT every data it deals with: the APML file, the ordered list of Web resources of interest and every user action used to define the relevance.

V. USAGE OF THE PROFILE

Searchy manages the profile in the same way it does with every resource shared in the DART Community, it can retrieve the UP from everywhere, independently from the device used, and at any time even if the node supposed to maintain an information is unreachable: the whole of the DART resource is redundantly stored.

A. Sorting query results

Searchy is aimed at offering search results according to user's needs: a key issue is to implement sorting functions to highlight the resources which are closest to user's preferences. As we said before, the User Profile and Web resources implement the same model, an ordered list of weighted categories, what allows Searchy to compare information ontologically different but formally equivalent.

The function we use to calculate the distance between user and resource is "Cosine Similarity" (6). This technique was primarily used in information retrieval for calculating similarity between two documents, where documents were usually represented as vectors of word frequencies and the similarity was the angle between the vectors. In our context,

[4] WordNet Domains Hierarchy: http://wndomains.itc.it/hierarchy.html

[5] A *synset* or synonym set is defined as a set of one or more synonyms that are interchangeable in some context without changing the truth value of the proposition in which they are embedded. http://en.wikipedia.org/

[6] "APML allows users to share their own personal Attention Profile in much the same way that OPML allows the exchange of reading lists between News Readers. The idea is to compress all forms of Attention Data into a portable file format containing a description of ranked user interests". http://www.apml.org/

categories weights replace words frequencies.

$$s(P,I) = \cos(P,I) = \frac{\vec{P} \cdot \vec{I}}{\|P\| * \|I\|} = \frac{\sum_{j \in C} P(c_j) \cdot I(c_j)}{\sqrt{\sum_{j \in C} \left(P(c_j)\right)^2} \cdot \sqrt{\sum_{j \in C} \left(I(c_j)\right)^2}} \quad (6)$$

where
$P(c_j)$ = weight of j-th category c in profile P
$I(c_j)$ = weight of j-th category c in item I
C= WordNet Domains categories set

B. Collaborative recommendation

Recommendation techniques have a number of possible classifications, as well expounded by Robin Burke[7]. Their differences are based on: (i) the kind of information that the system has before the recommendation process begins; (ii) the information that user must communicate to the system in order to generate a recommendation; and (iii) the algorithm that combines these data to arrive at its suggestions.

"Collaborative recommendation is probably the most familiar, most widely implemented and most mature of the technologies" [5] and so we chose to accept this technique. Typically, collaborative recommender systems aggregate ratings of items, recognize resemblance among users on the basis of their ratings, and generate new recommendations based on inter-user comparisons. User profile consists of a vector of items and their ratings.

The main difference, between our system and classical ones, is the user model. As said before, the profile consists of an ordered list of weighted categories, the same model used to describe Web resources and user's query string; this approach allows Searchy to involve in the recommendation process not only users who share the same rating vector with the user whom the prediction is for, but, above all, expert users that happen to have a profile similar to the query string.

Searchy identifies the Qualified Users Set (QUS) by the same approach used to calculate distance between user and resource, as explained before in the (6), and delegates it to perform the query result. In other words, Searchy identifies a set of users similar to the active user or similar to the query, then, among the resources that have high value of *uR* for these qualified users, looks for those resources which match the query. Finally, the result set, already filtered according to QUS' suggestions, is sorted as explained in the previous section "Sorting query results".

C. Query formulation and results management

Besides obtaining user's information and providing personalized results to user's queries, Searchy helps the user during the phase of query formulation. It allows user to modify his profile, both before to submit a query, so she/he can choose which QUS will be built, and after the results are displayed, so she/he can dynamically change their order.

[7] Recommendation techniques: collaborative, content-based, demographic, utility-based and knowledge-based. [5]

Moreover, Searchy provides the GUIs to perform semantic queries. The Semantic Module analyzes the query and returns the categories set related to the whole string, and above all, the possible meanings set related to each term, so the user can choose the right meaning for each word, and then she/he can submit, instead of a query by terms, a query by meanings. The returned categories can be used to add a filter on the result set. The users can select one or more categories and Searchy will hide every resource that doesn't contain almost one of the selected categories in its semantic description.

VI. The Implementation

Searchy is a Firefox extension[8], and it follows Mozilla standards, so it is cross-platform[9]: XUL[10] provides the interface, JavaScript handles the functionality.

After installing the extension on Firefox, Searchy can be activated by "Searchy" button in "Tools" menu, it can work in two ways: *Hidden* or *Sidebar*.

A. Hidden mode

In this modality Searchy is completely transparent to the user, but it is able to do every operation useful to learn user information. The extension uses the functionalities that XPCOM[11] interfaces offer, so it can observe the Browser's activities and handle its events, in this way Searchy can catch user's behaviours which it uses to calculate the user profile: every time user browses, puts into bookmarks list, downloads, prints or sends a page, Searchy transparently stores the event in the RDHT. Searchy communicates with the DART Node via AJAX[12] calls to the REST[13] services installed on the back-end. Moreover the assistant provides some keyboard shortcuts to perform *voting* and *banning* actions: ALT + number keys from 1 to 5 for voting and ALT+0 to ban the page.

B. Sidebar mode

This modality implements every *Hidden mode*'s functionalities but adds graphical interfaces and shows Searchy in Firefox sidebar. The assistant provides GUIs for:

1) Performing registration, login and logout;
2) Setting server options, necessary for connecting to a DART Node;

[8] "Extensions add new functionality to Mozilla applications such as Firefox and Thunderbird. They can add anything from a toolbar button to a completely new feature. They allow the application to be customized to fit the personal needs of each user if they need additional features, while keeping the applications small to download." http://developer.mozilla.org/en/docs/Extensions

[9] We are testing Searchy on Firefox 2.0.x installed both on Windows XP and on Linux Fedora/Ubuntu.

[10] XML User Interface Language: http://www.xulplanet.com/

[11] Cross Platform Component Object Model. It is a cross platform component model from Mozilla. http://www.mozilla.org/projects/xpcom

[12] Asynchronous JavaScript and XML [Ajax: A New Approach to Web Applications: www.adaptivepath.com/ideas/essays/archives/000385.php]

[13] Representational State Transfer

3) Filling demographic information[14];
4) Assisting user during query formulation, as previously said in section "Query formulation and results management" (Fig. 1);

5) Showing a list of URLs correlated to the current page, this list of URLs is dynamically created performing a simple query to DART; Searchy analyzes the viewed page, recognizes the most relevant words and uses them to formulate the query.
6) Showing the actions history and for managing it in order

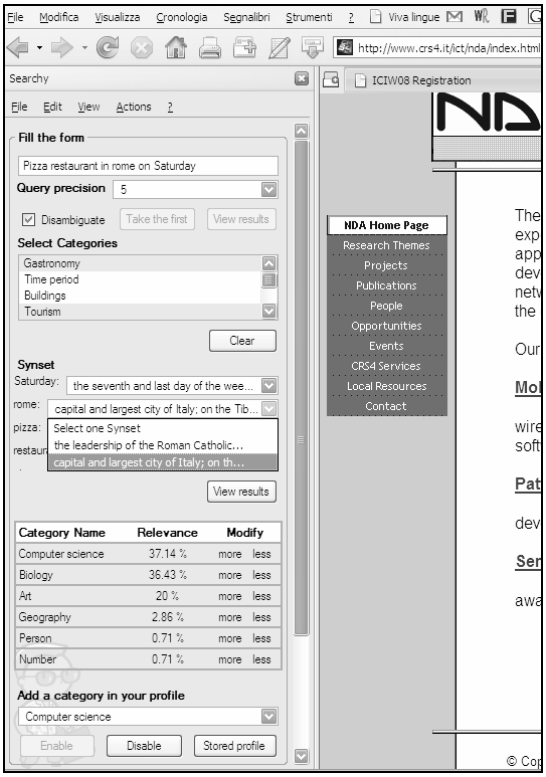

Fig. 1. Searchy's graphical user interface: Query Formulation (Sidebar mode).

to correct possible Searchy's misunderstandings.

C. Results presentation

The current results visualization is a simple list of Web resources, not different from what the other search engines offer; we are working on UI which can give a more intuitive association between user and resource.

Currently Searchy shows, for each resource: the page title; the description; the text portions where the terms or the synsets were found and the resource URL. In addition to the classical view, we add a numeric value to specify the closeness between user profile and resource and also two cake graphs to show it, we add a little page preview too. A map, which shows geographical points related to the resource, is available too. Thanks to GeoNames[15] services, the Semantic Module can enrich the resource meta information with the earth coordinates related to each name of place it finds in the page text.

D. Data protection

As we said before, Searchy, to identify similar users and to involve them in the activity of suggesting interesting sites, needs to get access to all the user profiles which are stored in the RDHT. To allow this feature we decided to keep these information freely available. The only solution we have adopted to protect the users from having their profiles accessed by unauthorized parties, is to keep every information anonymous. The user profiles and their own lists of interesting sites are identified in the RDHT by using the same name, surname and password used to perform the login. This identification data are client side processed by the MD5 algorithm before to be sent to the DART Node.
The above mentioned analysis, regarding the adoption and implementation of a privacy policy, we recognize is not really exhaustive. The lack of performance in this feature resides on the fact that the privacy polices and encrypting data are not a research target for the DART project. Consequentially, protecting information methods, when Searchy exchanges data with the node, is not a feature under analysis.

VII. Experimental results

We have successfully tested the behaviour of Searchy on different operative systems and different Firefox versions, at the moment. The agent, thanks to the cross-platform architecture of Mozilla Firefox and AJAX and REST standards, correctly sends to the DART Node all the data useful to learn the profile and properly manages the Node responses.

As we said before, Searchy's main goal is to identify interesting Web sites and, following the methodologies applied in many of the related works, we decided to evaluate this functionality involving the final users, in the next months we intend to focus our work in this direction. We planned to create two groups of ten users. Every member has an age, a cultural background and internet skills comparable to these of the other users of her/his own group, but completely different from these of the second group's members.

We decided to test the system for two weeks, to verify if Searchy is able to cluster users in the same two groups we previously defined. After the testing period we planned to submit users a survey to gather their opinions about the *uR* values automatically conferred to the viewed resources by Searchy. Users' responses are useful to understand if explicit

[14] At the moment Searchy locally stores demographic data but doesn't use them.

[15] GeoNames is a geographical data base freely available and accessible through various Web services, under a Creative Commons attribution license. The GeoNames database contains over 8,000,000 geographical names corresponding to over 6,500,000 unique features. http://www.geonames.org/

and implicit feedbacks' weight is being well estimated by the system and, above all, to understand above which *uR* value, a resource can be taken into account. For the moment, we consider as interesting for a user only the resources which are over her/his own average *uR* value.

VIII. Acknowledgment

The authors would like to thank the precious contribute of all the team of researchers who worked with them to the development of the architecture of the Dart project and the partners Tiscali S.p.A. and DIEE, Dept. of Electrical and Electronic Engineering, University of Cagliari.

IX. Future works and Conclusion

Our future work will mainly focus on four issues: conducing more experiments in order to evaluate the effectiveness of the system; monitoring user's behaviour on Mozilla Thunderbird too; improving results GUI; and applying the technologies shown in this paper on different fields, mainly on TV programming. We intend to install Searchy in a Media Center, so that it can learn user's TV tastes and automatically record TV shows in order to build a personalized and alternative channel.

In this paper we proposed Searchy, a software agent embedded in the Web browser and developed to assist users during Web searches. Observing user's behavior during browsing session, the agent identifies the relevant Web resources and, by content-based algorithms, deduces which are the interesting topics and builds the UP. Searchy orders and filters the results set according to UP and identifies similar users able to suggest interesting items.

References

[1] M. Angioni, et al., "DART: The Distributed Agent-Based Retrieval Toolkit" in *Proc. of CEA 07*, Gold Coast, Australia, 2007, pp. 425-433.

[2] M. Angioni, et al., "User Oriented Information Retrieval in a Collaborative and Context Aware Search Engine" in *Proc. of WSEAS Transactions on Computer Research*, vol. 2, Funchal, Portugal, 2007, pp. 79-86.

[3] M. Angioni, R. Demontis, F. Tuveri, "Enriching WordNet to Index and Retrieve Semantic Information" in *Proc. of 2nd International Conference on Metadata and Semantics Research*, Corfu, Greece, 2007.

[4] R. Armstrong, et al., "Webwatcher: A learning apprentice for the world wide web", in *1995 AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments*, March 1995, pp 6-12.

[5] R. Burke, "Hybrid Recommender Systems: Survey and Experiments" in *User Modeling and User-Adapted Interaction*, *Volume 12 , Issue 4,* Kluwer Academic Publishers Hingham, MA, USA, 2002, pp. 331-370.

[6] M. Claypool et al., "Implicit Interest Indicators", in *Proc. of ACM Intelligent User Interfaces Conference (IUI '01)*, Santa Fe, NM, USA, 2001, pp. 33-40.

[7] C. Chen, M. Chen, Y. Sun, "PVA: A Self-Adaptive Personal View Agent", *Journal of Intelligent Information Systems*, 2002, pp. 173-194.

[8] L. Chen, K. Sycara, "Webmate: A personal agent for browsing and searching" in *Proc. of 2nd International Conference on Autonomous Agents*, 1998, pp. 132-139.

[9] S. Dumais, et al., "Stuff I've Seen: A System for Personal Information Retrieval and Re-use", in *Proc. of SIGIR 2003*, Toronto, Canada, 2003.

[10] D. Kelly, J. Teevan, "Implicit Feedback for Inferring User Preference: A bibliography", *SIGIR Forum*, Vol. 37, No. 2, 2003, pp. 18-28

[11] D. Kelly, N. J. Belkin, "Reading Time, Scrolling and Interaction: Exploring Implicit Sources of User Preferences for Relevance Feedback" in *Proceedings of ACM Conference on Research and Development in Information Retrieval (SIGIR .01)*, New Orleans, LA, USA, 2001, pp. 408-409.

[12] J. Konstan, et al., "GroupLens: Applying Collaborative Filtering to Usenet News" *n `special section: recommendation systems' in CACM March 1997*, Vol. 40, No. 3, 1997, pp. 77-87.

[13] T. Kurki, et al., "Agents in delivering personalized content based on semantic metadata", *in Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, Stanford, USA, 1999, pp. 84-93.

[14] H. Liebermann, "Letizia: An Agent that assists web browsing", in *Proc. of the 13th Int. Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA, USA, 1995, pp. 924-929.

[15] B. Magnini et al., "The Role of Domain Information in Word Sense Disambiguation" in *Natural Language Engineering*, special issue on *Word Sense Disambiguation*, Vol. 8, No. 4, Cambridge University Press, 2002, pp. 359-373.

[16] B. Magnini, C. Strapparava, "User Modeling for News Web Sites with Word Sense Based Techniques" in *User Modeling and User-Adapted Interaction*, Vol. 14, No. 2, 2004, pp. 239-257.

[17] S. Middleton, D. De Roure, N. Shadbolt, "Capturing knowledge of user preferences: ontologies in recommender systems", *in Proc. of the 1st International Conference on Knowledge Capture (K-Cap2001)*, Victoria, BC, Canada, 2001

[18] Miller et al., *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, USA: C. Fellbaum, 1998.

[19] M. Morita, Y. Shinoda, "Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval", in *Proc. of ACM Conference on Research and Development in Information Retrieval (SIGIR .94)*, Dublin, Ireland. 1994, pp. 272-281.

[20] M. Pazzani, J. Muramatsu, D. Billsus, "Syskill & Webert: Identifying interesting web sites" in *Proc. of the Thirteenth National Conference Artificial Intelligence (AAAI)*, Portland, USA, 1996, pages 54-61.

[21] A. Pretschner, "Ontology based personalized search", Master's thesis, The University of Kansas, Lawrence, KS, USA, 1999.

[22] Y. W. Seo, B. T. Zhang, "Learning User's Preferences by Analyzing Web-Browsing Behaviors", in *Proc. of International Conference on Autonomous Agents*, Barcelona, Spain. 2000, pp. 381-387.

[23] D. Sleator, D. Temperley, "Parsing English with a Link Grammar" technical report CMU-CS-91-196, Department of Computer Science, Carnegie Mellon University, 1991. A version of this paper appears in *Proc. of the Third International Workshop on Parsing Technologies*, 1993, pp. 277 - 292.

[24] A. Soro, C Lai, "Range-capable Distributed Hash Tables", in *Proc. of Third International Workshop on Geographic Information Retrieval - GIR'06*, Seattle, USA, 2006, pp. 332–342.

[25] F. Tanudjaja, L. Mui, "Persona: A Contextualized and Personalized Web Search", in *Proc. of the 35 Annual Hawaii International Conference on System Sciences (HICSS'02)*, Vol. 3, Big Island, Hawaii, USA, 2002, pp. 1232-1240.

[26] C. Thomas, G. Fischer, "Using agents to personalize the web", in *Proc. ACM IUI'97*, Orlando, Florida, USA, 1997, pp. 53-60.

[27] R. W. White, I. Ruthven, J. M. Jose, "The Use of Implicit Evidence for Relevance Feedback in Web Retrieval" in *Proceedings of 24th BCS-IRSG European Colloquium on IR*, Glasgow, Scotland, 2002, pp. 93-109.

[28] G. Xu, Y. Zhang, X. Zhou, "Towards User Profiling for Web Recommendation", in *Proc. of Australian Conference on Artificial Intelligence*, 2005, pp. 415-424.