Web Intelligence and Agent Systems: An international journal 1 (2003) 219–234
IOS Press

Ontology-based personalized search and browsing[1]

Susan Gauch[a], Jason Chaffee[b] and Alexander Pretschner[c]
[a]*Department of EECS, 2001 Eaton Hall, University of Kansas, Lawrence, KS 66045, USA*
Tel.: +1 785 864 7755; Fax: +1 785 864 0387; E-mail: sgauch@ku.edu
[b]*MetaTV, Inc., Mill Valley, CA 94941, USA*
E-mail: jasonchaffee@metatv.com
[c]*Institut fur Informatik, Technische Universitat Munchen, Munchen, Germany*
E-mail: pretschn@in.tum.de

Abstract. As the number of Internet users and the number of accessible Web pages grows, it is becoming increasingly difficult for users to find documents that are relevant to their particular needs. Users must either browse through a large hierarchy of concepts to find the information for which they are looking or submit a query to a publicly available search engine and wade through hundreds of results, most of them irrelevant. The core of the problem is that whether the user is browsing or searching, whether they are an eighth grade student or a Nobel prize winner, the identical information is selected and it is presented the same way. In this paper, we report on research that adapts information navigation based on a user profile structured as a weighted concept hierarchy. A user may create his or her own concept hierarchy and use them for browsing Web sites. Or, the user profile may be created from a reference ontology by 'watching over the user's shoulder' while they browse. We show that these automatically created profiles reflect the user's interests quite well and they are able to produce moderate improvements when applied to search results.

Keywords: Ontologies, personalization, browsing, Web navigation, conceptual search

1. Introduction

The Web has experienced continuous growth since its creation. As of March 2002, the largest search engine contained approximately 968 million indexed pages in its database [54]. As the number of Internet users and the number of accessible Web pages grows, it is becoming increasingly difficult for users to find documents that are relevant to their particular needs. Users of the Internet basically have two ways to find the information for which they are looking: they can browse or they can search with a search engine. Browsing is usually done by clicking through a hierarchy of concepts, or *ontology*, until the area of interest has been reached. The corresponding node then provides the user with links to related Web sites. Search engines allow users to enter keywords to retrieve documents that contain these keywords. The browsing and searching algorithms are essentially the same for all users.

The ontologies that are used for browsing content at a Web site are generally different for each site that a user visits. Even if there are similarly named concepts in the ontology, they may contain different types of pages. Frequently, the same concepts will appear with different names and/or in different areas of the ontology. Not only are there differences between sites, but between users as well. One user may consider a certain topic to be an "Arts" topic, while a different user might consider the same topic to be a "Recreation" topic. Thus, although browsing provides a very simple mechanism for information navigation, it can be time

[1]This work was partially supported by NSF CAREER Award 97-00307 and NSF ITR 0225676 (SEEK).

consuming for users when they take the wrong paths through the ontology in search of information.

The alternate navigation strategy, search, has its own problems. Indeed, approximately one half of all retrieved documents have been reported to be irrelevant [5]. One of the main reasons for obtaining poor search results is that many words have multiple meanings [28]. For instance, two people searching for "wildcats" may be looking for two completely different things (wild animals and sports teams), yet they will get exactly the same results. It is highly unlikely that the millions of users with access to the Internet are so similar in their interests that one approach to browsing or searching, respectively, fits all needs. What is needed is a solution that will "personalize" the information selection and presentation for each user.

This paper explores the OBIWAN project's use of ontologies as the key to providing personalized information access. Our goals it to automatically create ontology-based user profiles based and use these profiles to personalize the results from an Internet search engine and to also use them to create personalized navigation hierarchies of remove Web sites. Section 2 provides an introduction to some related work and in Section 3 we describe the automatic creation of user profiles based on a user's browsing behavior. In Section 4, we show how these profiles can be used to improve search results, and in Section 5 we discuss how users can create their own profiles and use them as the basis for personalized browsing. We conclude by summarizing the results of these investigations and we discuss our current focus on conceptual, personalized search.

2. Related work

The following section presents related work on ontologies and personalization. Since we create our user profiles automatically using text classification techniques, we will also review research in this area.

2.1. Classification

Classification is one approach to handling large volumes of data. It attempts to organize information by classifying documents into the best matching concept(s) from a predefined set of concepts. Several methods for text classification have been developed, each with a different approach for comparing the new documents to the reference set. These include: comparisons between a variety of frequently-used vector representations of the documents (Support Vector Machines, k-nearest neighbor, linear least-squares fit, tf * idf); use of the joint probabilities of the words being in the same document (Naive Bayesian); decision trees; and neural networks. A thorough survey and comparison of such methods is presented in [43,49,62].

Classification has been applied to newsgroup articles, Web pages, and other online documents. The system described in [21] classifies NETNEWS articles into the best matching news groups. The implementation uses the vector space model to compare new articles to those articles manually associated with each news group. The system presented in [17] is based on a probabilistic description-oriented representation of Web pages, and a probabilistic interpretation of the k-nearest neighbor classifier. It takes into account: 1) features specific to Web pages (e.g., a term appears in a title, a term is highlighted), 2) features standard to text documents, such as the term frequency. The k-nearest neighbor approach has also been used by [32] in a system that uses classification techniques to automatically grade essays.

2.2. Ontologies

One increasingly popular way to structure information is through the use of ontologies, or graphs of concepts. One such system is *OntoSeek* [18], which is designed for content-based information retrieval from online yellow pages and product catalogs. OntoSeek uses simple conceptual graphs to represent queries and resource descriptions. The system uses the *Sensus* ontology [26], which comprises a simple taxonomic structure of approximately 70,000 nodes. The system presented in [30] uses *Yahoo*! [63] as an ontology. The system semantically annotates Web pages via the use of Yahoo! categories as descriptors of their content. The system uses *Telltale* [10,11,44] as its classifier. Telltale computes the similarity between documents using n-grams as index terms.

The ontologies used in the above examples use simple structured links between concepts. A richer and more powerful representation is provided by *SHOE* [20, 35]. SHOE is a set of Simple HTML Ontology Extensions that allow WWW authors to annotate their pages with semantic content expressed in terms of an ontology. SHOE provides the ability to define ontologies, create new ontologies which extend existing ontologies, and classify entities under an "is a" classification scheme.

2.3. Personalization

Personalization is a broad field of active research. Applications include personalized access to online information such as personalized "portals" to the Web, filtering/rating systems for electronic newspapers [9], Usenet news filtering, recommendation services for browsing, navigation, and search. Usenet news filtering systems include GroupLens [27], PSUN [56], NewT [55], *Alipes* [38], and SIFT [61]. SiteIF [57] and ifWeb [2] aim to provide personalized search and navigation support. InformationLens [37] is a tool for filtering and ranking e-mails. Implicit rating and filtering are, among other topics, discussed in [41] and [42]. Finally [58] describes a system for expertise location (Java source code) [46] describes approximately 45 personalization systems and contains a detailed bibliography.

Many personalization projects have focused on navigation. Syskill & Webert [43] also recommends interesting Web pages using explicit feedback. If the user rates some links on a page, Syskill & Webert can recommend other links on the page in which they might be interested. In addition, the system can construct a Lycos query and retrieve pages that might match a user's interest. *Wisconsin Adaptive Web Assistant* (WAWA) [52,53] also uses explicit user feedback to train neural networks to assist users during browsing.

Personal WebWatcher [38] is an individual system that is based on WebWatcher [1,23]. It "watches over the user's shoulder," but it avoids involving the user in its learning process because it does not ask the user for keywords or opinions about pages. *Letizia* [33,34] is a similar individual system that assists a user when browsing by suggesting links that might be of interest and are related to the page the user is currently visiting. The system relies on implicit feedback including links followed by the user or pages and/or bookmarked pages. *WebMate* [8] is an individual system based on a stand-alone proxy that can monitor a user's actions to automatically create a user profile. Then the user can enter an URL and WebMate will download the page, check for similarity with the user's profile, and recommend any similar pages. *Amalthaea* [40] is a server-based system that employs genetic algorithms to also try to identify Web pages of interest to users.

Most personalization systems are based on some type of user profile, most commonly a set of weighted keywords. Systems that use structured information rather than simple lists of keywords include PEA [39] and SiteSeer [48], both of which use bookmark information, PSUN [56] which uses K-lines, and SiteIF [57] which uses semantic networks. By incorporating temporal information [60] uses an extended user profile model that distinguishes between a user's short term and long term interests. Similar to our work, SmartPush [29] uses concept hierarchies for user profiles. In contrast, however, these are quite small (40–600 nodes), and weight adjustments are done using data that *explicitly* describes document contents. It is doubtful that handmade hierarchical content annotation of data will be done on a large scale, limiting the applicability of this approach.

In order to build a user profile, some source of information about the user must be collected. Commercial systems, e.g., MyYahoo, explicitly ask the user to provide information about themself which simply stored to create a profile. Explicit profile creation is not recommended because it places an additional burden on the user, the user may not accurately report their own interests, and the profile remains static whereas the user's interests may change over time. Thus, implicit profile creation based on observations of the user's actions is used in most recent projects [7] describes the types of information available. His model considers the frequency of visits to a page, the amount of time spent on the page, how recently a page was visited and whether or not the page was bookmarked. Similar to our research, the user's surfing behavior is used to create the user profiles in Anatagonomy [50], Letizia [33, 34], Krakatoa [24], Personal WebWatcher [38], and WBI [3].

Our user profiling technique differs from other approaches due to our focus on automatically creating user profiles based on ontologies. In our use of ontologies, we overlap somewhat with initiatives aimed at creating a Semantic Web [4]. In the Semantic Web world, the information encoded is given a well-defined meaning using predefined ontologies. The predefined ontologies, on the other hand, consist of term descriptions and their interrelationships that allow making inferences and retrieving more relevant information that the keyword based search. Systems such as Ontogator [22], an ontology-based image retrieval and recommendation browser, Ontobroker [13], a hyperbolic browsing tool that uses ontologies to annotate Web documents and answer queries, and OntoRama [12], a generic ontology viewer, make use of the latest tools and ontology representation formats. However, these proposals tend to focus on encoding semantics into the Web pages to describe their content, whereas we use classification techniques to automatically create profiles for users and/or Web sites. The Semantic Web approaches also differ in that they provide a mechanism for representing a wide variety of link types and/or link labels between concepts whereas our hierarchies are simpler since they handle only unlabelled links (assumed to represent parent-child relationships) between concepts.

In the Semantic Web approach, the ontologies are modeled using ontology representation languages such as the Extensible Markup Language (XML), the Resource Description Framework (RDF), RDF Schema, DAML+OIL, or the Web Ontology Language (OWL) [59]. Although using representation languages to define ontologies and then annotating the pages initially requires more work, it facilitates reasoning and inference on the defined information and promises to improve searching and browsing the Semantic Web. Our work may inter-operate with Semantic Web approaches by automatically classifying documents (or users) with respect to concepts within an ontology, helping to automate the ontological markup of web pages or the creation of ontologically-based user profiles.

3. Automatic creation of user profiles

In our system, the user profile is created automatically and implicitly while the users browse. The user profile is essentially a reference ontology in which each concept has a weight indicating the user's perceived interest in that concept. Profiles are generated by analyzing the surfing behavior of the user, specifically the content, length, and time spent on each Web page they visit. The Web pages the user visits are automatically classified into the concepts contained in the reference ontology and the results of the classification are accumulated. This causes the concepts in the reference ontology to receive weights based on the amount of related information the user has browsed. No user feedback is necessary.

3.1. Reference ontology

Since our user profile is essentially a weighted ontology, our first goal was to locate or create a reference ontology on which to base our user profile. Rather than create our own ontology, a time consuming process, we chose to base our ontology on already existing subject hierarchies. Online portals such as Yahoo.com [Yahoo 2003], and About.com [About 2003], provide manually-created online subject hierarchies and a set of Web pages manually associated with each subject designed to organize Web content for easy browsing by end-users.

One of the advantages of our approach is that our system can work a reference ontology created from any subject hierarchy that has associated textual information. To date, we have based our reference ontology on subject hierarchies and associated Web pages from Yahoo, Magellan, Lycos, and the Open Directory Project. Since most subject hierarchies allow a given subject to have more than one parent, the subject "hierarchy" is actually a directed acyclic graph (DAG). We create a pure hierarchy out of the DAG by replicating the subject (and associated Web pages) in each location to which it is linked. The subject hierarchies typically contain well over 100,000 subjects arranged in a DAG with a depth exceeding 10. Since we wish to create a relatively concise user profile that identifies the general areas of a user's interests, we create our reference ontology by using concepts from only the top levels of the subject hierarchy. In addition, since we want concepts that are related by a broader-narrower relationship, we remove subjects that were linked based on non-conceptual criteria, e.g., alphabetic or geographic associations.

The automatic profile creation described in the next section is based upon text classification algorithms. In order to classify the user's Web pages into concepts in the reference ontology, we require a collection of associated Web pages to be used as training data for the classifier. Thus, we exclude concepts from the reference ontology if there are too few Web pages to adequately train the classifier.

For the experiments on personalized search described in Section 4, the reference ontology on which the user profile is based consisted of the 4.417 concepts from the top four levels of the subject hierarchy created by Magellan [Magellan 1999] that had adequate training data. For the experiments in personalized browsing described in Section 5, the reference ontology contained 5.863 concepts from the top four levels of Lycos's [Lycos 1999] subject hierarchy. To create each reference ontology, we spidered the Web site to create a local copy of their subject hierarchy. We also parsed each subject page to identify and collect the content Web pages linked for each subject.

3.2. User profile creation

The user profiles are created by periodically processing the user's Web cache to extract the urls of Web pages that they visited. A spider collects the identified Web pages and then the pages are classified into the appropriate concept(s) in the reference ontology using a vector-space classifier.

3.2.1. Training the classifier

The Web pages manually linked to each concept by the creator of the subject hierarchy were used as training data for the classifier. The same number of content documents were used to train each concept. The content documents for each concept were concatenated to create a collection of super-documents, D. The super-documents were pre-processed to remove high-frequency function words (stopwords) and HTML tags. Finally, the Porter stemmer [14] was used to reduce each word to its root to decrease the effect of word variations on the classification.

In a vector-space classifier, each concept j is represented by a vector, c_j, containing one entry per term in the vocabulary. The weight for a given term is a factor of the frequency of the

$$tc_{ij} = tf_{ij} * idf_i \tag{1}$$

term in the super-document for the concept, tf_{ij}, and the rarity of that term in other concepts, idf_j. In more detail, the tc_{ij}, weight of term i concept j is given by:

where

$D =$ the collection of super-documents

$t_i =$ the ith term in the vocabulary

$d_j =$ the jth super-document

$$tf_{ij} = \text{number of occurences of } t_i \text{ in } d_i \tag{2}$$

$$idf_i = \log \left(\frac{\text{number of documents in D}}{\text{number of documents in D that contain } t_i} \right) \tag{3}$$

The dimensionality of the concept vectors is very large, one dimension for every word used in any document in the collection. This dimensionality is somewhat reduced by removing stopwords and further reduced by stemming. However, since most super-documents contain only a small fraction of the possible words, and absent terms receive a weight of 0, these concept vectors are very sparse. Because not all documents were the same length, the concepts vary somewhat in the amount of training data. To compensate for this, the term weights in each concept vector are normalized by the vector magnitude, creating unit length vectors. Thus, ntc_{ij}, the normalized weight of term i concept j is given by:

$$ntc_{ij} = \frac{tc_{ij}}{\text{vector} - \text{length}_j} \tag{1}$$

where

$$\text{vector} - \text{length}_j = \sum_i tc_{ij} \tag{4}$$

We have trained the classifier with a wide variety of Web pages per concept and have found that the classification algorithm is not particularly sensitive to the amount of training data. Using a variety of subject hierarchies, we found that anywhere from 5 pages to 60 pages per concept can provide reasonably accurate classification. When trained with 10 documents per category, the amount of training data used in the experiments reported here, the correct concept for a document was the top-ranked concept 51% of the time and occurred among the top 5 ranked concepts 75% of the time.

3.2.2. Building the user profile

The Web pages collected from each user's Web browser cache folder were periodically classified into the appropriate concept(s) in the reference ontology. For each of the visited pages, a document vector was calculated using the same formulae used for the concept vectors. The similarity between the vector for document k, d_k, and the vector associated with concept j, c_j, was calculated using the cosine similarity measure (see Eq. (5)). The concepts with the highest similarity values were assumed to be those most related to the content of the surfed page.

$$\text{similarity}(d_k, c_j) = \frac{\sum_{i=1}^{n}(ntd_{ik} * ntc_{ij})}{\sqrt{\sum_{i=1}^{n} ntd_{ik}^2 * \sum_{i=1}^{n} ntc_{ij}^2}} \tag{5}$$

where

$ntd_{ik} =$ the normalized weight of the ith term in document k

$ntc_{ij} =$ the normalized weight of the ith term in concept j

$n =$ the number of unique terms in the document collectionD

Initially, a user's profile starts off with all concepts in the ontology having a weight of zero. As pages are classified with respect to the reference ontology, the values reported by the classifier are added to the top five concept's weights. Over time, as more and more pages are classified, the weights are accumulated. Concepts into which many visited documents are classified continue to increase in weight, and it is our hypothesis that higher weighted concepts represent concepts of greater user interest.

We also investigated the influence of two other factors in document-concept similarity calculation: the duration of the visit and the page length. Intuitively, if a user spends a long time on the page, their interest value in that page should be increased. However, if the page is long, the influence of the time factor should possibly be decreased since the increased time may be due to the amount of information presented, not the level of interest. We used four different formulae combining time and page length factors (see Eq. 6) to adjust the strengths of the similarity between a browsed document and the concepts.

$$\begin{aligned}\mathrm{similarity}(d_k, c_j) &= \mathrm{timelengthfactor}\\ &\quad *\mathrm{similarity}(d_k, c_j)\end{aligned} \tag{6}$$

where

timelengthfactor is calculated in one of four ways:

$$\frac{\mathrm{time}}{\mathrm{length}} \quad (i)$$

$$\log \frac{\mathrm{time}}{\mathrm{lenght}} \quad (ii)$$

$$\log \frac{\mathrm{time}}{\log \mathrm{length}} \quad (iii)$$

$$\log \frac{\mathrm{time}}{\log \log \mathrm{lenght}} \quad (iv)$$

$\mathrm{time} =$ the amount of time the user spent visiting the page in seconds

$\mathrm{lenght} =$ the length of the page in bytes

The first *timelength* factor formula, (i), is a straightforward normalization of the time spent browsing the page by the length of the page. The other three formulae decrease the effect of the *timelength* factor on the overall similarity value by applying a log to the result. For formulae (iii) through (iv), the importance of length as a normalizing factor on time is decreased by first logging and then log-logging the length component.

The evaluation of our user profile creation algorithm consisted of two parts [47]. First, we tested our profile creation algorithm to determine whether or not it was able to create a stable user profile. The second experiment validated our automatically generated user profiles against actual user interests.

3.3. Profile convergence

One would assume that each person has a relatively stable collection of interests that may change over time [31]. We wished to determine how long it takes our system to identify this core set of interests. In our work, a user profile is said to be convergent if the number of concepts with non-zero interest values converges over time. Users varied in the number of categories to which their profiles converged, most containing between 50 and 200 concepts that account for 95% of the total accumulated profile weight. Low-weighted concepts were ignored to filter "noise" introduced by text classification and/or user navigation errors.

For the experiments, a group of sixteen users were monitored for 26 days. These sixteen users together surfed 7,664 documents. The users spent a mean of 54.6 seconds per page, with a standard deviation of 93.4 seconds. 20% of all pages were visited for less than 5 seconds. All profiles showed a tendency to converge after roughly 320 pages, or 17 days, of surfing when the document-concept similarity values were adjusted by the *timelength* formulae (iii) and (iv). These are the formulae that minimize the effect of the length of the page on the document/concept similarity calculation. However, when we used *timelength* formulae (i) and (ii), the profiles did not converge. This indicates that the length of a surfed page is not an important factor when calculating the user's interest in a particular page (and thus their interest in the page's associated concepts). Thus, it seems that users can tell at a glance that a page is irrelevant and, in general, reject it quickly regardless of its length.

3.4. Comparison with actual user interests

Although convergence is a desirable property, it does not measure the accuracy of the generated profiles. Thus, the sixteen users were shown the top twenty concepts in their profiles in random order and asked how appropriately these inferred concepts reflected their true interests. For both the top ten and top twenty concepts, approximately one half of the concepts represented actual interests (5.2 and 10.5 respectively), one quarter represented errors and the remaining quarter represented topics of marginal interest. Bearing in mind that the "good" concepts have been chosen out of 4.400 concepts, this result is encouragingly accurate. 75% of the twenty categories chosen reflect actual interests even though these represent only 0.5% of all possible concepts.

4. Personalized search results

Because queries are so short, search engines generally do not receive enough detail about the user's information need. As a result, many retrieved documents are irrelevant. Although the profiles created as described in Section 3 were not perfect, we hypothesized that they were accurate enough to allow a search engine to provide personalized search. We evaluated the use of our automatically created user profiles for personalized search using two different approaches:

1) *Re-ranking* Re-ranking algorithms apply a function to the document-query match values and/or the rank orders returned by the search engine. If that function is well chosen, it should move relevant documents higher in the list and demote non-relevant documents.
2) *Filtering* Filtering systems determine which documents in the result sets are relevant and which are not. Good filters remove many non-relevant documents and preserve the relevant ones in the results set.

4.1. Evaluation

For a given query, re-ranking was done by modifying the ranking that was returned by the ProFusion metasearch engine [45]. We classified each of the documents in the result set into the categories of the reference ontology. Rather than using the full documents, which would require a serious delay while the documents were fetch, we classified only the titles and summaries shown on the search engine result page. These, according to [5] and [43], are sufficient for accurate classification. We then wanted to estimate the user's interest in the document by examining the user's interests in the concepts to which the document belonged. This was done by averaging the user profile's values for the four concepts identified as being the most similar to the document.

Once we had an estimate for the user's interest in the document's concepts, we re-calculated the match values between the query and the documents. For each search engine result r, we calculated new match values, new_wtr, based on the match value returned by the search engine, the similarity between the result and its top concepts, and the level of user interest in the top concepts. Formula 7 shows the recalculation of the document/query match weight:

$$\text{new\_wt}_r = wt_r * (0.5 + \frac{1}{4}\sum_{i=1}^{4} u_{crl}) \qquad (7)$$

where

wt_r is the weight returned by the search

engine for result r

u_{crl} is the user's interest in concept c_{rl}

(from their profile)

c_{rl} is the lth most highly weighted

concept for result r

To compare the results produced by the different reranking formulae, we used the eleven point precision average [19]. The eleven point precision average evaluates ranking performance in terms of *recall* and *precision*. Recall is a measure of the ability of the system to present all relevant items (i.e., it is the percentage of relevant documents retrieved), and precision is a measure of the ability of a system to present only relevant items (i.e., it is the percentage of retrieved document that are relevant).

Sixteen users were each asked to submit three queries (48 total). Two queries per user were used for training (32 total) and the third query was reserved for evaluation (16 total). The results were presented in random order, and the users were asked to judge each result as being "relevant" or "non-relevant."

On average, before re-ranking, only 8.7 of the twenty retrieved pages were considered to be relevant. This is consistent with the findings in [5] which reports that roughly 50% of documents retrieved by search engines are irrelevant. The reranking of documents by promoting those that classify into concepts of high interest to the user produced an overall performance increase of 8% (see Fig. 1). In particular, the biggest improvement is occurs within the most highly-ranked documents. Since the top documents are those most likely to be examined by a user, improvement at the top of the list is encouraging.

We also evaluated the ability of the user profile to filter documents from the result set. After calculating per-

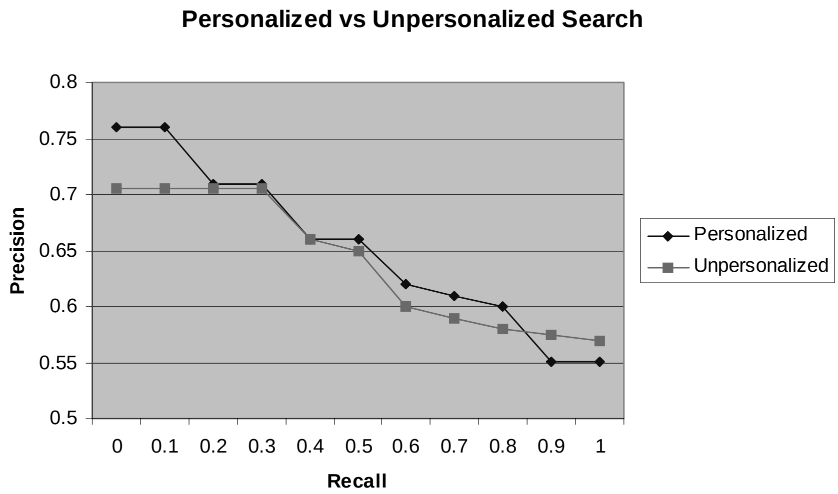

Fig. 1. Recall and prescision with and without personalized reranking.

sonalized match values (see Formula 7), we excluded documents whose revised match values fell below a threshold. We evaluated a variety of threshold values and achieved approximately a 2:1 ratio of irrelevant documents removed to relevant documents removed at all values of the threshold. Clearly, as the threshold was raised, more documents of both types were removed.

4.2. Discussion

We were able to create large, structured, user profiles entirely automatically by classifying content from the user's browsing cache into concepts in a reference ontology. These profiles were shown to converge and to reflect actual user interests quite well. To evaluate their usability, two applications have been investigated: re-ranking and filtering search results. In terms of re-ranking, performance increases of up to 8% were achieved. In terms of filtering, roughly a 2:1 ratio of irrelevant documents to relevant documents were removed.

5. Personalized browsing

Most Web sites present a hierarchy of links to help users to navigate through the online content. However, each site uses their own criteria to develop site-specific arrangements of content. This requires users to learn how navigate each site. Our goal with *Ontology Based Informing Web Agent Navigation* (*OBIWAN*) [64] system was to use classification to arrange the content from a variety of Web sites into a single reference ontology. Thus, each site could be navigated using the same concept-based arrangement of content regardless of how the online information was linked by the original site designers.

OBIWAN.s Local Characterizing Agent (LCA) [64] spiders and classifies the Web pages of a site using a reference ontology derived from the ontology used by Lycos [36]. The classification algorithm for the spidered web pages is the same one used for the creation of user profiles (see Formula 5). Essentially, we create a profile for a Web site in the same way we create a profile for a user, with three major differences. First, for user profiles, we collect the representative pages by processing the user's browsing cache whereas for Web sites we collect the representative pages by spidering content linked from the home page. Second, we adjust the similarity weights for user pages during classification based on the length of time they spent viewing the page and this is not applicable for Web site content. Finally, we stored only the accumulated weight for each concept in the user profile whereas, with Web sites, we also record the top matching concept for each classified Web page so that the pages can later be presented as when that concept is browsed.

Just like Yahoo, or any other hierarchically arranged Web site, the user can browse OBIWAN's arrangement of the content of a site by clicking up and down a hierarchy of concepts. With OBIWAN's Local Browsing Agent (LBA), however, all sites appear to be organized conceptually according to a reference ontology, even though each site may have originally been designed around a different hierarchy or arranged non-conceptually (e.g., alphabetically). Once several sites have their contents classified into the same ontology, OBIWAN's Regional Browsing Agent (RBA) can be used to simultaneously browse multiple Web sites.

Figure 2 shows a screen shot of the RBA. The left frame displays the reference ontology. The stars associated with each concept indicate the amount of information classified into that concept, accumulated over all the sites in the region. In this example, there are three sites being browsed simultaneously: the University of Stanford; the Kansas City Star newspaper; and CNN's Sports Illustrated. The current concept being browsed by the user is "DSL," a sub-concept of "Data Communications." The right frame lists the sites in the region that have information classified into that concept and, once again, the stars beside the site names indicate the amount of information available for that concept at each site. Stanford has much more information on DSL than the others. By clicking on the site name, the user could launch the Local Browsing Agent (LBA) and be able to see the actual web pages classified into that concept for the chosen Web site.

The work reported here extends OBIWAN's Local Browsing Agent so that each site can browsed using the user's own ontology rather than the single system-wide reference ontology. To create a personal ontology, a user amasses a collection of Web pages that he or she arranges into a hierarchy based on his or her worldview. The system then finds a mapping from the reference ontology concepts to concepts in the personal ontology. Using this mapping, the user can browse any site that has been characterized by OBIWAN with his or her personal ontology without reclassifying the documents. Since OBIWAN will characterize every site in the same manner, and each user's personal ontology reflects their view of the world, they will be able to browse Web pages in a personalized, consistent manner.

5.1. System architecture

Each concept in an ontology needs a set of documents that were manually assigned to that concept. For the reference ontology, these documents were collected from the Lycos site. For the personal ontology, the sample documents are provided by the user.

The personal browsing system needs to map from reference ontology concepts to the best matching concept in the personal ontology. To do this, it must calculate the similarity between each concept in the reference ontology and the concepts in the personal ontology. Figure 3 shows the system architecture for the personalized browsing system.

5.2. Mapping the reference ontology to the personal ontology

Each user submits their personal ontology, a hierarchical tree of concepts that represents their view of the world. For our experiments, the tree was required to contain at least ten concepts with at least five sample pages for each concept. The goal of the mapping phase is to map every concept in the reference ontology to a concept in the personal ontology. However, since personal ontologies tend to be much smaller and more narrowly focused than the reference ontology, many concepts will remain unmapped. Thus, we augment the personal tree with an extra concept called "All-Others" to hold the concepts from the reference ontology that do not map to a corresponding concept in the personal ontology.

We take a multi-phase approach to mapping from each reference ontology concept to the best matching personal ontology concept. While it is possible for a reference ontology concept to map to multiple personal ontology concepts, this would indicate that the personal concepts are more fine-grained than the reference concepts. Since our reference ontology is very large (5.863 concepts), this is not likely to occur. Thus, we simplified our mapping algorithm to focus on mapping each reference concept to the best matching, single personal concept. In practice, our users tended to create concepts that were at least as broad or broader than the reference concept.

The first step maps from the personal ontology concepts to the reference ontology concepts. As described in Section 3.2.1, the classifier is trained and vectors are created for each concept in the reference ontology. The same technique is used to create vectors for each concept in the personal ontology. The similarity between the personal concept vector and reference ontology vector is calculated using the cosine similarity measure (Formula 5) and the top 30 matches are returned. The result of this process is a one-to-many mapping from personal ontology concepts to reference ontology concepts.

After the first step, the same reference ontology concept may appear on multiple lists (i.e., be mapped to more than one personal ontology concept). So, the next step filters the results of step 1 to identify the best matching personal concept for each reference concept. This produces a one-to-one mapping from reference ontology concepts to personal ontology concepts.

The first two steps map individual reference ontology concepts to their best matching personal ontology con-

Fig. 2. OBIWAN's regional browsing agent.

cept. Since the personal ontology concepts tend to be broader in scope than the reference ontology, we next map any unmapped descendents of reference ontology mapped nodes to the same personal ontology concept as their nearest ancestor. Where an unmapped node has multiple mapped ancestors at the same level, the mapping with the highest weight is chosen. This has the effect of mapping entire subtrees rather than just individual concepts. For instance, in Fig. 4 it can be seen that the concept "Anime" has ancestors "Animation" and "Arts-&-Entertainment", with "Animation" being the closest ancestor. Therefore, "Anime" has two possible ancestors to which it could be mapped.

After the system has mapped a reference ontology concept to a personal ontology concept, a mapping factor is calculated which measures the closeness of the match between the concepts. This factor is normalized by the sizes of the mapped concepts and the value of the reference concept's term vector matched against itself (see Formula 8). The mapping factor can be viewed as a measure of our confidence in the mapping.

$$\text{mapping factor} = \frac{\dfrac{\text{matching weight}}{\text{file size of personalized concept}}}{\dfrac{\text{weight of reference concept queried against itself}}{\text{file size of reference concept}}} \qquad (8)$$

The mappings between reference ontology concepts and their top-matching personal ontology concepts, along with the mapping factor for each match, are stored in a mapping file.

5.3. *Mapping a site to the personal ontology*

Once the mapping file has been created, any site that has had its Web pages spidered and classified into the reference ontology concepts can easily be mapped to the personal ontology. If several concepts in the reference ontology map to one concept in the personal ontology, they are all merged together under the personal concept. If a concept in the reference ontology does not map to any concept in the personal ontology, the pages will remain in the reference ontology concept and be displayed in the "Other" concept of the personal ontology. Next, for pages in mapped concepts, the similarity value between the page and its reference ontology

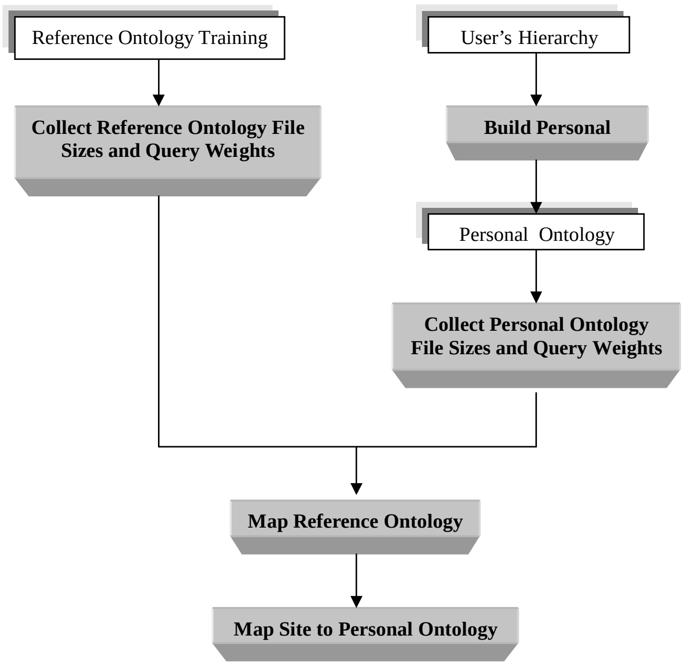

Fig. 3. Personalized browsing architecture.

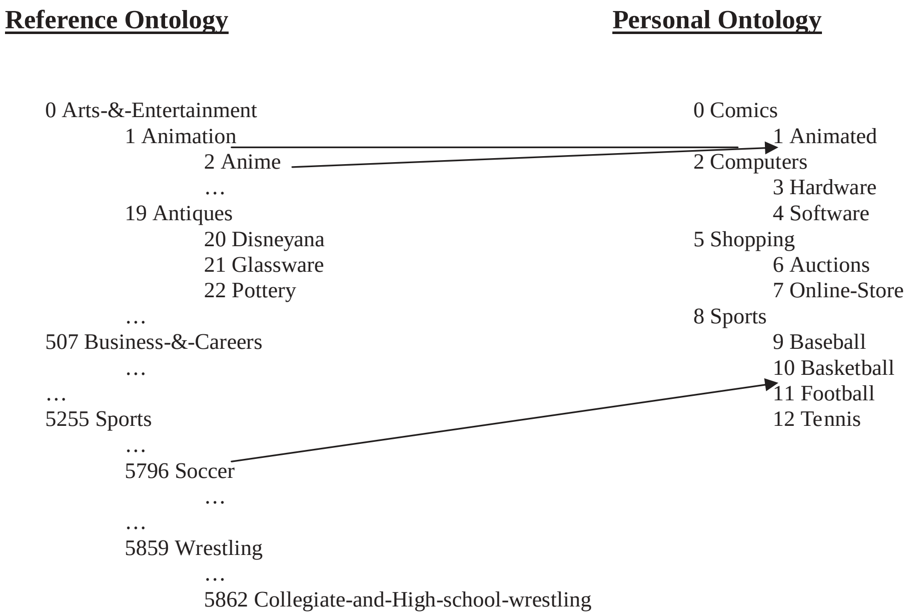

Fig. 4. The reference ontology mapped to a personal ontology.

concept is adjusted by multiplying it by the mapping factor between the reference ontology concept and the personal concept (see Formula 9).

$$\text{new weight} = \text{similarity between page and reference ontology concept} * \text{mapping factor} \quad (9)$$

After all pages have been mapped and their weights recalculated, the weights for the concept as a whole are calculated as the sums of the weights of mapped pages

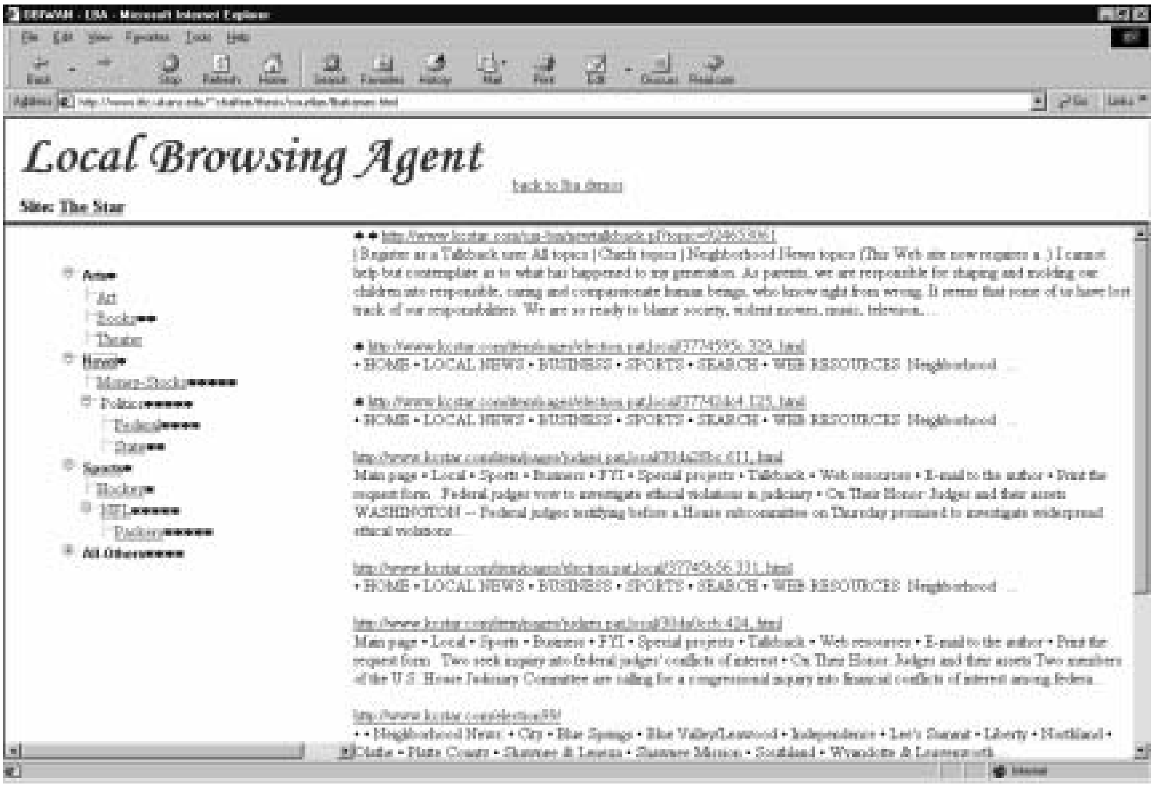

Fig. 5. Screen shot of a web site's content displayed after being mapped to the personal ontology.

plus the weight of any subtrees. Now the site can have its content browsed using the personal ontology rather than OBIWAN's reference ontology (see Fig. 5).

5.4. Evaluation

The system was evaluated by having five users create personal ontologies. Each user was asked to provide feedback on two different experiments. The first experiment asked each user to compare the reference ontology concept that was mapped to their personal concept and decide if it was mapped correctly. The second experiment had each user browse a site's Web pages after they had been mapped to their personal ontology. Each user reported whether or not each page that was mapped to their personal ontology was mapped to the correct concept.

5.4.1. Evaluating ontology mappings

The user was given a Web interface to view each one of their concepts and every concept from the reference ontology that had been mapped to the personal concept. Also, the user was able to view the training data from the reference ontology concepts. The user was asked to give a Yes/No answer to the question of whether or not the reference ontology concept matched the personal ontology concept.

We then used the user responses to determine a threshold. We expected that the percentage of correct mappings would increase if we eliminated mappings below some threshold. When the threshold is increased, the number of concepts that are mapped both correctly and incorrectly is reduced. In the extreme, if the threshold is set to 100%, there are no results because there are no mappings. Therefore, another measure was used to measure "correctness" for each threshold (see Formula 10). Table 1 shows the precision, recall, and correctness values for each threshold value. We found that a threshold of 0.3 produced the highest number of correct mappings.

$$\begin{aligned} \mathrm{correctnes\ s} = {} & (\text{number kept are correct} + \\ & \text{numberdrooped that are} \\ & \mathrm{incorrect})/(\text{total number of} \\ & \mathrm{concepts}\text{mapped with no} \\ & \mathrm{threshold}) \end{aligned} \tag{6}$$

5.4.2. Evaluating site mappings

The evaluation of the ontology mappings showed that a threshold of 0.3 produced the highest value for correctness. Therefore, each user's concept mappings were pruned using a threshold of 0.3 before an individual site's Web pages were mapped to their personal ontologies. Only the top ten mapped pages were kept for any concept in the personal ontology (see Fig. 5). As with the previous experiment, the user was asked to give a Yes/No answer on whether or not each page that had been mapped to a personal concept belonged there.

We then used the user responses to determine a threshold for the mapping weight of an individual page. We expected that the percentage of correct mappings would increase if we eliminated mappings below some threshold.

5.5. Discussion

We evaluated the system with two measures, precision and correctness. Precision measures the number of correct pages that were seen vs. the total number of pages that were seen. Correctness measures the number of correct pages seen plus the number incorrect pages not seen vs. the total number mapped.

It was found that the concepts mapped correctly with a precision of 49% and correctness of 49% with no threshold. The best results were achieved with a mapping threshold of 0.3. This produced a precision of 53% and a correctness of 55%. Using a threshold for mapping concepts will reduce the number of reference concepts that actually are mapped, but it will cause the concepts that are mapped to have a higher relevance with the personal concepts. There are several factors that affected the results. First, the concepts that were submitted by the users were not always conceptual in nature, e.g., a user's name. Second, the training data in both the reference ontology and the personal ontologies was not as good as we expected. Although we had what appeared to be an adequate number of pages, many of the pages contained very little content, or the content included a template that added noise to the frequency statistics of words.

We found that individual pages mapped correctly with a precision and correctness of 50% with no threshold. In contrast to the concept mappings, the use of a threshold did not improve precision or correctness. We believe the main source of the low correctness was primarily due to errors introduced when the Web site pages were mapped to the reference ontology concepts rather than when the reference ontology concepts were mapped to the personal ontology concepts.

Currently, the user is asked to provide an ontology for the system. Most users do not want to take the time to create an ontology, especially one that only contains concepts. Therefore, a system that creates the ontology for the user would be beneficial. Finally, the system as described maps from a reference ontology to a personal ontology. It could also be used to map between two commonly found ontologies on the Web. For example, Yahoo!'s ontology could be used as the reference ontology and Lycos' ontology could be the ontology the system will map to. Then, a user could browse Yahoo!'s categories with the Lycos ontology.

6. Conclusions and future work

This paper reviews extensions to the OBIWAN project that are working towards the goal of personalized navigation of online information. Our research revolves around using weighted ontologies to represent users and/or Web content conceptually. The general approach begins with the use of a reference ontology created automatically by spidering any of a number of online subject hierarchies. The Web pages linked within each subject are spidered and used as training data for a text classifier. Ample and accurate training data for each concept, and a reliable and robust classification algorithm, is key to the success of this approach.

By classifying the contents of a user's browsing cache into the reference ontology concepts, we are able to automatically create user profiles. Our user profiles are unique in that they are weighted ontologies rather than the more frequently used feature vectors. After approximately 320 pages per users, these profiles converged to a stable set of 50 to 100 concepts. We found that incorporating the time spent browsing the page into the classification formula lead to profile convergence, but that the size of the page was not an important factor. We were able to demonstrate that the user profiles were reasonably accurate in that 75% of the concepts identified were judged by the users to accurately reflected their actual interests.

We created a personalized search system that made use of the automatically created user profiles. Documents in the result set of an Internet search engine were classified based on their titles and summaries. Those documents that were classified into concepts that were highly weighted in the user's profile were promoted by a re-ranking algorithm. Overall, an 8% improvement

Table 1
The effect of varying thresholds on concept mapping accuracy

Mapping factor threshold	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Precision	49%	49%	49%	53%	52%	45%	34%	35%	36%	100%
Recall	100%	100%	99%	84%	41%	16%	5%	2%	1%	0%
Correctness	49%	49%	49%	55%	53%	49%	49%	50%	51%	51%
Mapped Correctly (seen)*	585	585	577	491	241	91	29	11	4	1
Mapped Correctly (not seen)**	0	0	8	94	344	494	556	574	581	584
Total Seen***	1192	1192	1179	931	460	202	85	31	11	1

* All concepts or pages which were mapped correctly and were not removed due to the threshold.
** All concepts or pages which were mapped correctly and were removed due to the threshold.
*** All concepts or pages that were mapped and were not removed due to the threshold.

Table 2
The effect of varying thresholds on page mapping accuracy

Mapping weight threshold	0	100	200	300	400	500	600	700	800	900
Precision	50%	50%	50%	46%	37%	25%	19%	20%	21%	15%
Recall	100%	100%	82%	52%	29%	14%	9%	8%	7%	4%
Correctness	50%	50%	50%	45%	41%	36%	36%	38%	41%	42%
Mapped Correctly (seen)*	136	136	111	71	39	19	12	11	9	5
Mapped Correctly (not seen)**	0	0	25	65	97	117	124	125	127	131
Total Seen***	274	273	222	156	105	76	64	56	43	33

in the top 20 precision resulted from this personalized re-ranking, with the biggest improvement seen in the top-ranked results. The personalized search results reported here are promising, but they exposed two areas of possible improvement. First, the quality of the results is affected by the quality of the classification of documents into concepts that, in turn, is affected by the quality of the training data for each concept. Second, working as a post-process on the search results limits the ability of the system to achieve dramatic gains in search performance. If few of the twenty documents returned by the search engine address the user's information needs, then re-ranking and/or filtering cannot help.

In addition to personalized search, we investigated the use of classification techniques to map between user-created ontologies and the reference ontology to provide personalized browsing. OBIWAN's Local and Regional Browsing agents allow users to browse Web sites with respect to a consistent conceptual arrangement of the world. Web sites have their contents spidered and classified with respect to the reference ontology after which the reference ontology can be used to browse the spidered Web pages. By mapping from the user's own ontology to the reference ontology, users get a consistent arrangement of content that matches their own world view rather than the system's. Five users created their own ontologies and provided sample Web pages as training data. We were able to map from the personalized concepts to the reference ontology concepts and then use these mappings to browse Web sites that were pre-mapped into the reference ontology.

The current focus of our work is on improving personalized search results. The goal of our ongoing KeyConcept project [25] is to integrate the conceptual matching between the user's profile and the document concepts into the retrieval process itself. We have developed the first version of KeyConcept, a conceptual search engine that classifies documents as part of the indexing process. It allows users to input queries that contain keywords as well as concepts of interest and documents are retrieved based on combination of the keyword and conceptual similarity. Currently, the concepts are either explicitly entered by the user or inferred from ancillary text. This system was evaluated on a large collection and a significant increase in top ten precision was found. Our next step is to merge the automatically created user profiles with KeyConcept so that the user profile is implicitly submitted along with the query terms. Documents that match the supplied keywords and also the concepts in the user profile will be preferentially retrieved. It is our hope that we will thereby make a major step towards truly personalized search.

References

[1] Robert Armstrong, Dayne Freitag, Thorsten Joachims and Tom Mitchell, WebWatcher: A Learning Apprentice For The World Wide Web, in: *Proceedings of the AAAI Spring Symposium On Information Gathering*, 1995, pp. 6–12.

[2] F. Asnicar and C. Tasso, *ifWeb: A Protoype of User Model-Based Intelligent Agent for Documentation Filtering and Navigation in the World WideWeb*, in Proceedings of the 6th International Conference on User Modeling, June 1997.

[3] R. Barrett, P. Maglio and D. Kellem, *How to Personalize the Web*, in Proceedings of ACM CHI'97, Atlanta, USA, 1997.

[4] Tim Berners-Lee, James Hendler and Ora Lassila, The Semantic Web, *In Scientific American* **284**(5) (May 2001), 34–43.

[5] E. Casasola, *ProFusion Personal Assistant: An Agent for Personalized Information Filtering on the WWW,* Master's thesis, The University of Kansas, 1998

[6] Jason Chaffee and Susan Gauch, *Personal Ontologies For Web Navigation*, in Proceedings of the 9th International Conference On Information Knowledge Management (CIKM), 2000, pp. 227–234.

[7] Philip Chan, Constructing Web User Profiles: A Non-Invasive Learning Approach, in: *Web Usage Analysis and User Profiling, LNAI 1836, Springer-Verlag*, 2000, pp. 39–55.

[8] L. Chen and K. Sycara, *A Personal Agent for Browsing and Searching*, in Proceedings of the 2nd International Conference on Autonomous Agents, 1998, pp. 132–139.

[9] P. Chesnais, M. Mucklo and J. Sheena, The Fishwrap Personalized News System, in: *Proceedings of IEEE 2nd International Workshop on Community Networking: Integrating Multimedia Services to the Home*, Princeton, NJ, June 1995.

[10] G. Chower and C. Nicholas, Resource Selection in Café: an Architecture for Networked Information Retrieval, in: *Proceedings of SIGIR'96 Workshop on Networked Information Retrieval*, Zurich, 1996.

[11] G. Chower and C. Nicholas, *Meta-Data for Distributed Text Retrieval*, in Proceedings of First IEEE Metadata Conference, 1996.

[12] P. Eklund, S. Green and N. Roberts, Ontorama: Browsing an RDF ontology using a hyperbolic-like browser, in: *The First International Symposium on CyberWorlds* (*CW2002*), *Theory and Pracitces*, IEEE Press, November 2002.

[13] S. Decker, M. Erdmann, D. Fensel and R. Studer, Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information, in: *Database Semantics: Semantic Issues in Multimedia Systems*, R. Meersman et al., eds, Kluwer Academic Publisher, 1999, pp. 351–369.

[14] W.B. Frakes and R. Baeza-Yates, Information Retrieval: Data Structures and Algorithms, Prentice Hall, 1992.

[15] J. Goecks and J. Shavlik, Automatically Labeling Web Pages Based on Normal User Actions, in: *Proceedings of the Workshop on Machine Learning for Information Filtering, 1999 International Joint Conference on Artificial Intelligence* (*IJCAI*), Stockholm, Sweden, 1999.

[16] J. Goecks and J. Shavlik, *Learning Users' Interests by Unobtrusively Observing Their Normal Behavior*, In Proceedings of the 2000 International Conference on Intelligent User Interfaces, New Orleans, LA, 2000. pp. 129–132.

[17] N. Göver, M. Lalmas and N. Fuhr, *A Probabilistic Description-Oriented Approach for Categorising Web Documents*, in Proceedings of the 8th International Conference on Information and Knowledge Management, 1999, pp. 475–482.

[18] N. Guarino, C. Masolo and G. Vetere, OntoSeek: Content-Based Access to the Web, *IEEE Intelligent Systems* **14**(3) (May 1999), 70–80.

[19] D. Harman, *Evaluation Techniques and Measures*, in Proceedings of the 4th Text REtrieval Conference (TREC-4), 1996, pp. A6–A14.

[20] Jeff Heflin, James Hendler and Sean Luke, *SHOE: A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078* (*UMIACS TR-99-71*), University of Maryland at College Park, 1999. http://www.cs.umd.edu/projects/plus/SHOE/pubs/techrpt99.pdf.

[21] Wen-Lin Hsu and Sheau-Dong Lang, *Classification Algorithms for NETNEWS Articles*, in Proceedings of the 8th International Conference on Information and Knowledge Management, 1999, pp. 114–121.

[22] E. Hyvönen, S. Saarela and K. Viljanen, *Intelligent Image Retrieval and Browsing Using Semantic Web Techniques – A Case Study*, Presented at the International SEPIA Conference at the Finnish Museum of Photography, Helsinki, September 2003.

[23] T. Joachims, D. Freitag and T. Mitchell, *WebWatcher: A Tour Guide for the World Wide Web*, in Proceedings of IJCAI'97, August, 1997.

[24] T. Kamba, K. Bharat and M. Albers, *The Krakatoa Chronicle – An Interactive, Personalized Newspaper on the Web*, in Proceedings of the 4th International WWW Conference, 1995, pp. 159–170.

[25] KeyConcept Project, http://www.ittc.ku.edu/keyconcept.

[26] K. Knight and S. Luk, *Building a Large Knowledge Base for Machine Translation*, in Proceedings of American Association of Artificial Intelligence Conference (AAAI), 1999, pp. 773–778.

[27] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon and J. Riedl, GroupLens: Applying Collaborative Filtering To Usenet News, *Communications of the ACM* **40**(3) (March 1997), 77–87.

[28] Robert Krovetz and Bruce W. Croft, Lexical Ambiguity and Information Retrieval, *ACM Transactions on Information Systems* **10**(2) (April 1992), 115–141.

[29] T. Kurki, S. Jokela, R. Sulonen and M. Turpeinen, Agents in Delivering Personalized Content Based on Semantic Metadata, in: *Proceedings of the 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, 1999, pp. 84–93.

[30] Yannis Labrou and Tim Finin, *Yahoo! As An Ontology – Using Yahoo! Categories To Describe Documents*, in Proceedings of the 8th International Conference On Information Knowledge Management (CIKM), 1999, pp. 180–187.

[31] W. Lam, S. Mukhopadhyay, J. Mostafa and M. Palakal, Detection of Shifts in User Interests for Personalized Information Filtering, in: *Proceedings of ACM SIGIR'96, Zurich*, Switzerland, 1996.

[32] Leah S. Larkey, *Automatic Essay Grading Using Text Categorization Techniques*, in Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998, pp. 90-95.

[33] Henry Lieberman, *Letizia: An Agent That Assists Web Browsing*, in Proceedings of the 14th International Joint Conference On Artificial Intelligence, 1995, pp. 924–929.

[34] Henry Lieberman, *Autonomous Interface Agents*, In Proceedings of the ACM Conference on Computers and Human Interaction (CHI'97), May 1997.

[35] S. Luke, L. Spector, D. Rager and J. Hendler, *Ontology-Based Web Agents*, In Proceedings of the First International Conference on Autonomous Agents (AA'97), 1997.

[36] Lycos, http://www.lycos.com.

[37] T. Malone, K. Grant, F. Turbak, S. Brobst and M. Cohen, Intelligent Information Sharing Systems, *Communications of the ACM* (30) (May 1987), 390–402.

[38] Dunja Mladeni? Personal WebWatcher: Design and Implementation. Technical Report IJS-DP-7472, J. Stefan Institute, Department for Intelligent Systems, Ljubljana, Slovenia, 1998.

[39] M. Montebello, W. Gray and S. Hurley, A Personable Evolvable Advisor for WWW Knowledge-Based Systems, in: *Pro-*

ceedings of the 1998 International Database Engineering and Application Symposium (*IDEAS'98*), July 1998, pp. 224–233.

[40] Alexandros Moukas. Amalthaea: Information Discovery And Filtering Using A Multiagent Evolving Ecosystem. In Proceedings of the Conference on the Practical Application of Intelligent Agents and MultiAgent Technology, 1996, http://moux.www.media.mit.edu/people/moux/papers/PAAM96.

[41] D. Nichols, Implicit Rating and Filtering, in: *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*, November 1997.

[42] D. Oard and G. Marchionini, *A Conceptual Framework for Text Filtering*, Technical Report EE-TR-96-25 CAR-TR-830 CLIS-TR-9602 CS-TR-3643, University of Maryland, May 1996.

[43] Michael Pazzani, Jack Muramatsu, *Daniel Billsus, Syskill & Webert: Identifying Interesting Web Sites*, in Proceedings of the 13th National Conference On Artificial Intelligence, 1996, pp. 54–61.

[44] Claudia Pearce, Ethan Miller, *The TellTale dynamic hypertext environment: Approaches to scalability*, In Advances in Intelligent Hypertext, Lecture Notes in Computer Science, Springer-Verlag, 1997.

[45] ProFusion, http://www.profusion.com.

[46] Alexander Pretschner, *Ontology Based Personalized Search*, Master's thesis, University of Kansas, June 1999.

[47] Alexander Pretschner, Susan Gauch. Ontology Based Personalized Search. In Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), November 1999, pp. 391–398.

[48] James Rucker, Marcos J. Polanco, Siteseer: Personalized Navigation For The Web, *Communications of the ACM* **40**(3) (March 1997), 73–75.

[49] Miguel Ruiz, Padmini Srinivasan, *Hierarchical Neural Networks For Text Categorization*, in Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 1999, pp. 281–282.

[50] H. Sakagami and T. Kamba, *Learning Personal Preferences on Online Newspaper Articles From User Behaviors*, in Proceedings of the 6th International WWW Conference, 1997, pp. 291–300.

[51] G. Salton, *Automatic Text Processing*, Addison-Wesley, 1989. ISBN 0-201-12227-8.

[52] J. Shavlik and T. Eliassi-Rad, Intelligent Agents for Web-Based Tasks: An Advice-Taking Approach, in: *Working Notes of the AAAI/ICML-98 Workshop on Learning for text categorization*, Madison, WI, 1998.

[53] J. Shavlik, S. Calcari, T. Eliassi-Rad and J. Solock, *An Instructable, Adaptive Interface for Discovering and Monitoring Information on the World Wide Web*, in Proceedings of the 1999 International Conference on Intelligent User Interfaces, Redondo Beach, CA, 1999.

[54] Search Engine Showdown: Size statistics. http://www.searchengineshowdown.com/stats/sizeest.shtml.

[55] B. Sheth, *A Learning Approach to Personalized Information Filtering*, Master's thesis, Massachusetts Institute of Technology, February 1994.

[56] H. Sorensen and M. McElligott, PSUN: A Profiling System for Usenet News, in: *Proceedings of CIKM'95 Workshop on Intelligent Information Agents*, December 1995.

[57] A. Stefani and C. Strappavara, Personalizing Access to Web Sites: The SiteIF Project, in: *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98*, June 1998.

[58] A. Vivacqua, Agents for Expertise Location, in: *Proceedings of the 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, 1999, pp. 9–13.

[59] http://www.w3.org/2001/sw/WebOnt/.

[60] D.H. Widyantoro, T.R. Ioerger and J. Yen, Learning User Interest Dynamics with a Three-Descriptor Representation, *Journal of the American Society of Information Science and Technology* (*JASIST*) **52**(3) (2001), 212–225.

[61] T. Yan and H. García-Molina, *SIFT – A Tool for Wide-Area Information Dissemination*, in Proceedings of USENIX Technical Conference, 1995, pp. 177–186.

[62] Yiming Yang, Xin Liu, *A Re-Examination Of Text Categorization Methods*, in Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 1999, pp. 42–49.

[63] Yahoo! http://www.yahoo.com.

[64] Xiaolan Zhu, *Susan Gauch, Lutz Gerhard, Nicholas Kral, Alexander Pretschner, Ontology-Based Web Site Mapping For Information Exploration*, in Proceedings of the 8th International Conference On Information Knowledge Management (CIKM), 1999, pp. 188–194.