Using Personal Ontologies to Browse the Web

ABSTRACT

The publicly indexable Web contains an estimated 4 billion pages, however it is estimated that the largest search engine contains only 1.5 billion of these pages. 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. Often users must browse through a large hierarchy of categories to find the information for which they are looking. To provide the user with the most useful information in the least amount of time, we need a system that uses each user's view of the world for classification. This paper explores a way to use a user's personal arrangement of concepts to navigate the Web. This system is built by using the characterizations for a particular site created by the Ontology Based Informing Web Agent Navigation (OBIWAN) system and mapping from them to the user's personal ontologies. OBIWAN allows users to explore multiple sites via the same browsing hierarchy. This paper extends OBIWAN to allow users to explore multiple sites via their own browsing hierarchy. The mapping of the reference ontology to the personal ontology is shown to have a promising level of correctness and precision.

Keywords

Obiwan, ontology, ontologies, personalization, classification, browsing, Web navigation.

1. INTRODUCTION

Users of the Internet basically have two ways to find the information for which they are looking: they can search with a search engine, or they can browse. The publicly indexable Web contains an estimated 4 billion pages and it is estimated that the largest search engine contains only 1.5 billion of these pages, as of December 2001 [2]. 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. What is needed is a solution that will "personalize" the results for each user. Earlier work has focused on personalizing search results [31] while this research will focus on personalizing users' browsing experiences.

One problem with search engines is that the collection of documents is so huge that most queries return too many irrelevant documents for the user to sort through. It has been reported that approximately one half of all retrieved documents are irrelevant [3]. Browsing has many of the same problems that plague search engines. The ontologies that are used for browsing are generally different for each site a user visits, and even if there are similar concepts in the hierarchy often pages categorized under "Arts" on one site will not be the same type of pages categorized under "Arts" on a different site. 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. Also, unlike searching which brings together information from many sites, browsing can usually be done only one site at a time.

This paper demonstrates that it is possible to use a user's personal concepts to navigate the Web. This system is built by using the characterizations for a particular site created by the *Ontology Based Informing Web Agent Navigation (OBIWAN)* [4] system and mapping from them to the user's personal ontologies. OBIWAN will classify the Web pages of a site using a reference ontology based on the ontology used by *Lycos* [5]. Over a period of time, a user will amass a collection of Web pages that he or she will then arrange into a personal ontology based on his or her concepts of where each particular page belongs. The system will then try to determine the mapping of the reference ontology to the personal ontology. Using this mapping, the user can then 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 will be their concept of the world, they will be able to browse Web pages in a consistent manner.

2. SYSTEM OVERVIEW

The personal ontology system involves four phases: building a personal tree, collecting personal data, mapping the reference ontology to the personal ontology [7], and presenting the user with OBIWAN characterized sites via the personal ontology. Currently, the collection of data is a manual process. The users can accomplish this in a variety of ways. They can use the organization of their bookmarks as a personal ontology, or they can create a personal ontology from scratch and then collect documents they feel belong to each concept, or they can collect documents and arrange them to form an ontology.

During the mapping phase, all pages that are associated with one of the personal concepts are joined and treated as one *superdocument*. This superdocument is then compared with the superdocuments for each concept in the OBIWAN ontology [4] to identify the best matches. Then, we try to map each node in the OBIWAN ontology to a node in the personal ontology. Finally, the system provides a browsing ability to display an OBIWAN site using the personal ontology.

3. RELATED WORK

The related work is presented in three basic categories: personalization, classification, and ontologies. The following will discuss the different systems in each of the categories and relate them to OBIWAN and the system this paper describes.

3.1 Personalization

WebWatcher [8, 9] is a collaborative system that accompanies the user as he or she browses the Web. Basically, the user provides a few keywords describing a search goal and WebWatcher recommends related hyperlinks. It has been compared to a museum tour guide, as it interactively suggests where to go next. *Syskill & Webert* [13] 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.

Personal WebWatcher [10] is an individual system that is based on WebWatcher. It "watches over the user's shoulder" in a similar manner WebWatcher does, but it avoids involving the user in its learning process because it doesn't ask the user for keywords or opinions about pages. *Letizia* [15, 16] 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* [11] 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* [12] is a server-based system that employs genetic algorithms to also try to identify Web pages of interest to users. *Alipes* [14], on the other hand, attempts to gather and filter news articles on behalf of the user.

The previously mentioned systems are attempting to "person alize" the information on the Web to provide each user with more relevant information. All of the systems recommend online information to the user using keyword-based user profiles. Whereas this paper and OBIWAN use similar keyword vectors to represent individual concepts, our user profiles are weighted vectors of concepts, which are richer.

3.2 Classification

Classification is one approach to handling large volumes of data. It attempts to organize information by classifying, or categorizing, documents into the best matching category in a predefined set of categories. Classification has been applied to newsgroup articles, Web pages, and other online documents.

The system developed by Hsu and Lang [17] 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 developed by Gőver, Lalmas, and Fuhr [18] 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 Larkey [20] in a system that uses classification techniques to automatically grade essays. In contrast, Matsuda's and Fukushima's system [19] introduces the concept of document types, and attempts to classify Web pages into these document types. The system anticipates the classification of Web pages into document types according to the pages' structural characteristics.

3.3 Ontologies

One increasingly popular way to structure information is through the use of ontologies, graphs of concepts. One such system is *OntoSeek* [22], 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 [23], which comprises a simple taxonomic structure of about 50,000 nodes. The system developed by Labrou and Finin [24] uses *Yahoo!* [25] as an ontology. The system semantically annotates Web pages via the use of Yahoo! categories as descriptors of their content. The system uses *Telltale* [26, 27, 28] 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* [29, 30]. 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.

4. OBIWAN

OBIWAN [4] employs distributed intelligent agents to organize information on the Web. Each web site has local agents that characterize and provide access to the information at that particular site. These local agents communicate with regional agents that characterize and provide access to the information for a particular region of the Web.

4.1 Local Ontology Agent (LOA)

The *Local Ontology Agent (LOA)* encapsulates an ontology that represents the concepts contained in a Web site. The ontology consists of a hierarchy of subjects, or concepts. Currently, the LOA contains the Lycos categories, their inter-relationship and five to ten Web pages linked to each category to use as training data. Some of Lycos' categories are alphabetic listings; the LOA does not include these due to the fact that a letter is not a concept. The current reference ontology contains 5863 concepts and has a tree depth of four.

4.2 Local Characterizing Agent (LCA)

The *Local Characterizing Agent (LCA)* is given the ontology that has been created by the LOA and classifies each Web page from the local site into the most similar concept in the ontology using the vector space model. The superdocuments formed by concatenating the training documents collected for each concept are indexed to facilitate fast classification of new Web pages. Each Web page is attached to only the top-matching concept in the ontology. The weights of all the documents that match a particular concept are then accumulated for that concept. Next, all weights are propagated up the ontology. This means that any given concept's weight in the hierarchy is an accumulation of all its children's weights as well as its own weight.

4.3 Local Browsing Agent (LBA)

The *Local Browsing Agent (LBA)* guides users to reach Web pages of interest to them at a local site. It uses the reference ontology to provide consistent content-based browsing of many different sites. Visually, the LBA is similar to Microsoft's Window Explorer. The frame on the left of the screen contains the reference ontology and the frame on the right of the screen displays the Web pages from that site that have been classified into that particular concept. A relative weighting scheme is used to display the amount of content in each concept. Thus, the user can see how much information a site has in concepts of their own interest before they bother to click through to the actual category. The total weight of each concept is divided by the total weight of all sibling concepts. This relative weight is then used to assign from zero stars, little content compared to siblings, to five stars, a lot of content compared to siblings.

4.4 Regional Characterizing Agent (RCA)

Similar to the LCA, the *Regional Characterizing Agent (RCA)* maps entire sites to the reference ontology concepts. For every site in a region, the RCA will classify a site based on the mappings and values reported by the LCA characterized site. To accomplish this, the RCA simply merges the LCA results for all sites in a given region. The weights from all of the sites for each concept are accumulated to give each concept in the ontology its regional weight. Then, similar to the LCA, the weights are accumulated throughout the hierarchy of the ontology. This means that any given concept's weight in the hierarchy is an accumulation of all of its children's weights as well as its own weight.

4.5 Regional Browsing Agent (RBA)

The *Regional Browsing Agent (RBA),* which can be seen in figure 1, allows users to browse all of the sites in a region simultaneously. The RBA has the same appearance of the LBA with one exception: instead of displaying individual Web pages for each concept, it displays the names of sites in the region that contain content for that concept. Also, the RBA allows for a seamless transition to a particulars site's LBA results for a concept. As with the LBA, a relative weighting scheme is used to display the amount of content in each concept. The total weight of each concept is divided by the total weight of all sibling concepts. This relative weight is then used to assign anywhere from zero stars, little content compared to siblings, to five stars, a lot of content compared to siblings.

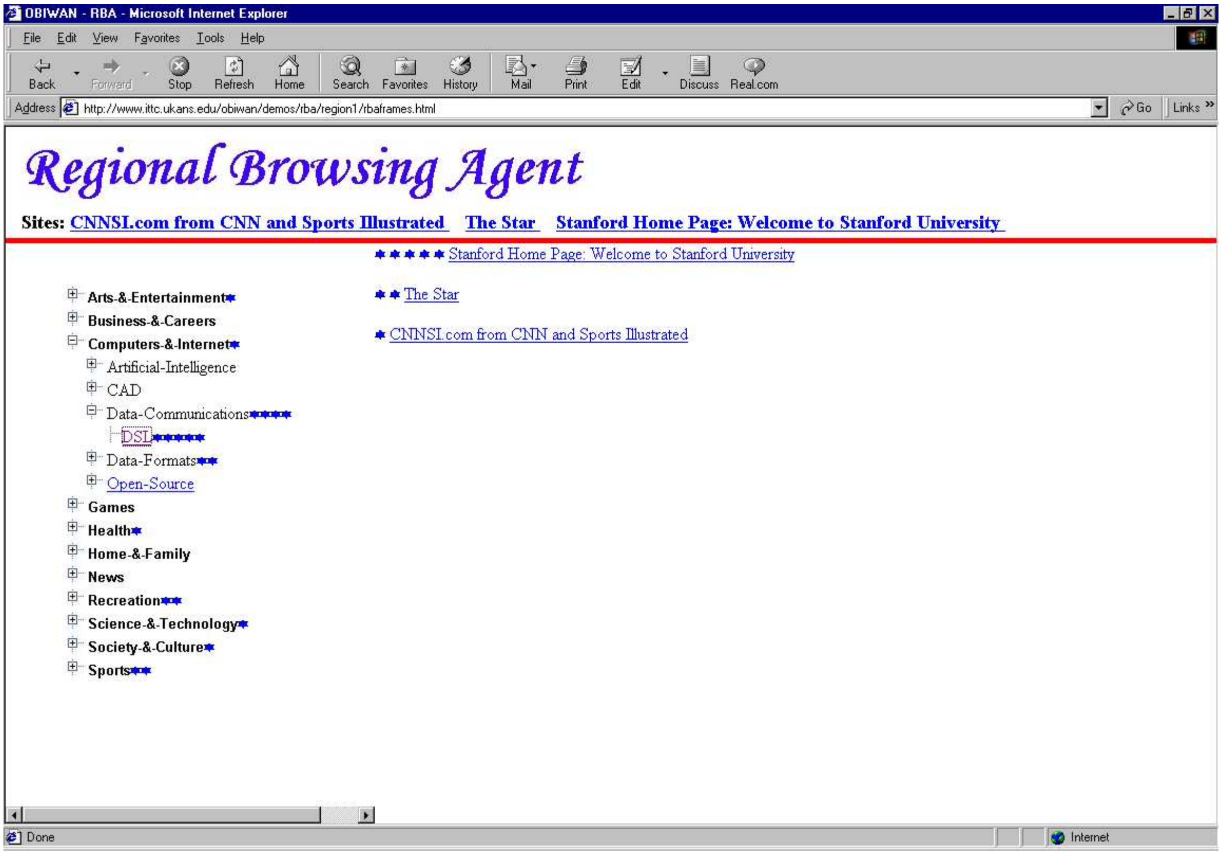

Figure 1. Regional Browsing Agent (RBA).

5. IMPLEMENTATION

The personal ontology system needs to map from reference ontology concepts to the best matching concept in the personal ontology. To do this, it must calculate the match value between each superdocument in the personal ontology with superdocuments in the reference ontology. The raw value of each match is normalized by the size of the superdocuments involved. This process can be broken into five distinct steps. First, the size of the superdocument for each concept in the reference ontology and the weight of the superdocument queried against itself needs to be calculated. Second, the personal ontology needs to be created for each user. Next, the size of the superdocument for each concept in the personal ontology and the weight of the superdocument queried against the reference ontology training data needs to be calculated. Then, the mapping from the reference ontology to the personal ontology is calculated. Finally, a particular site is mapped to the personal ontology. Figure 2 shows the system architecture

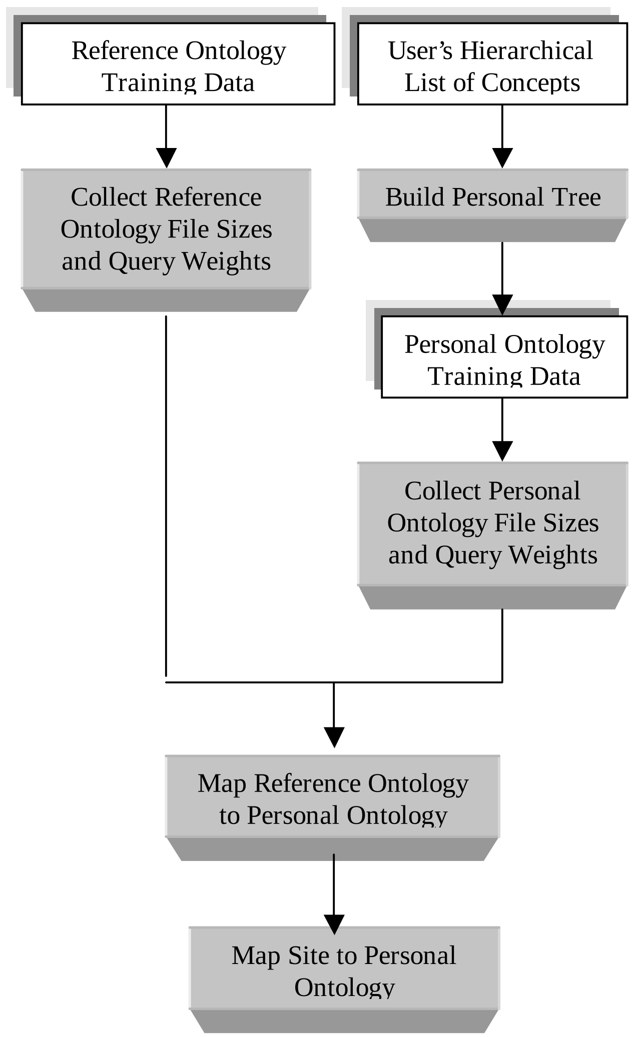

Figure 2. System architecture.

5.1 Calculating the Reference Ontology's Concept Sizes and Weights

The first process in mapping a reference ontology to a personal ontology is to calculate the size of the superdocument for each concept in the reference ontology and the weight when the superdocument is queried against itself. This process needs to be done only once. The results can be reused for each user.

The first task is to tokenize the superdocument, ignoring HTML tags and stopwords and stemming the words using the Porter stemmer [6]. Then, the size of the pages is calculated by using the total frequency tf_{ij} of each token t_i in the superdocument d_j multiplied by the inverse document frequency idf_I for term t_i which is calculated using the entire collection of superdocuments D.

Finally, the superdocument is classified by the LCA. The superdocument is queried against itself to give the weight of an exact match in our index of concepts. The value of the similarity of the superdocument to its associated concept is used as a baseline for comparing the quality of other matches to the concept. The baseline needs to be calculated for each superdocument because pivoted normalization [33] is used with our vector space model.

5.2 Building the Personal Ontology

The second step builds a personal ontology. Currently, this is done by having the user submit a hierarchical tree of concepts that represents their view of the world. Each user was required to submit a tree with at least ten nodes and at least five pages in each node. Next, this tree is augmented with an extra concept called "All-Others" to hold the concepts from the reference ontology that do not map to the personal ontology. The personal training data must be collected for the personal ontology. A Web interface was set up that allows users to select a concept in their ontology and submit URLs of Web pages they believed should belong to those concepts. All leaves in the personal tree are required to have a minimum of five pages and a maximum of ten pages associated with it. A node is not required to have any pages associated with it, but if the user decides to classify documents into the nodes, then they will need to have a minimum of five pages and a maximum of ten pages. Once the training data is collected, the sizes and weights can be calculated. As with the reference ontology, each concept's documents are tokenized into one superdocument whose weight is calculated as before.

5.3 Mapping the Reference Ontology to the Personal Ontology

The goal of the mapper is to map every concept from the reference ontology to the personal ontology. This goal is unlikely to be fully achieved, which is why the "All-Others" category was created. Every concept that is not mapped remains in the reference ontology and is placed under the "All-Others" category. As the first step mapping from the reference ontology to the personal ontology, the superdocument for each personal concept is matched against the collection of reference superdocuments. Again, the vector space model is used. At run time, it is decided how many top matches will be returned (currently 30). The result of this process is a many-to-many mapping between personal and reference ontology concepts.

Next, the system looks for the best inverse mappings from reference ontology concepts to personal concepts to produce a many-to-one mapping from reference ontology concepts to personal ontology concepts. If a reference ontology concept is mapped to more than one personal ontology concept, only the highest weighted mapping is kept.

When a reference ontology concept is mapped, we consider that as mapping the entire subtree of which that concept is the root. We next process the mappings once more to identify all unmapped descendents of mapped nodes and map those reference ontology concepts 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. For instance, in Figure 3 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.

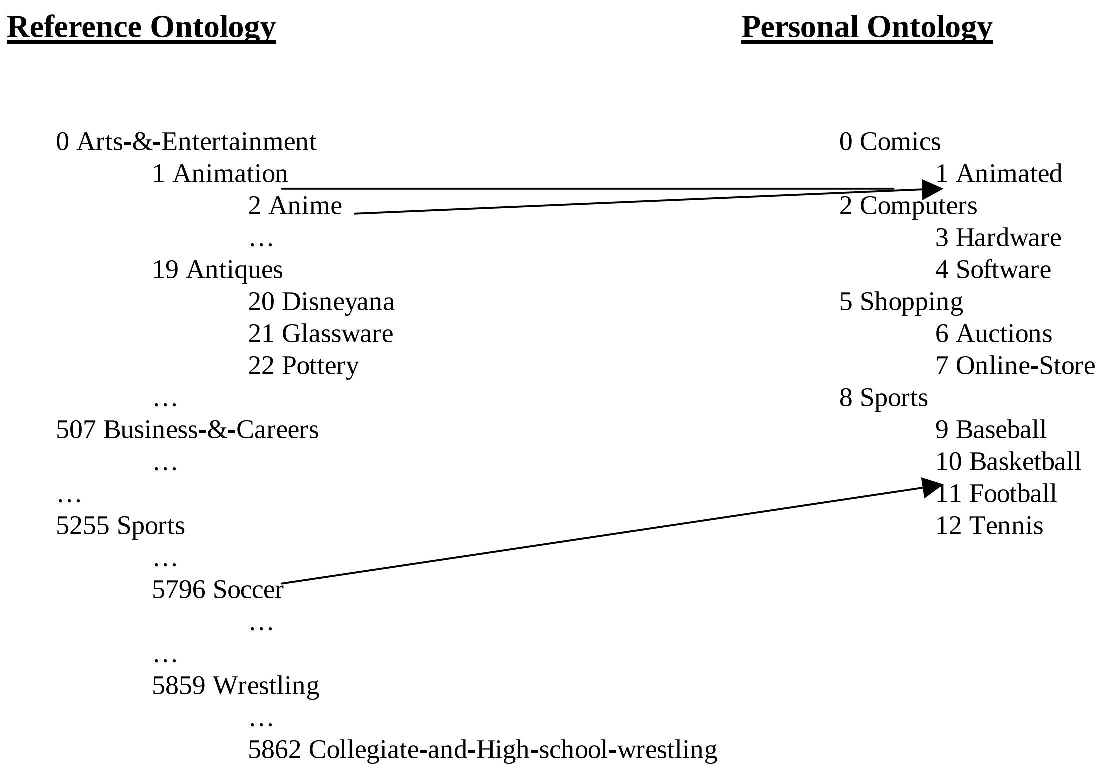

Figure 3. Mapping of reference ontology to personal ontology.

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 normalized by the sizes of the mapped concepts and the value of the concept’s superdocument matched against itself.

5.4 Mapping a Site to the Personal Ontology

Once the mapping file has been created, any site that has been characterized with the LCA 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 doesn’t map to any concept in the personal ontology, the pages will remain in the reference ontology concept. Next, the weights must be recalculated for each page that is mapped to the personal ontology.

The new weight is calculated by using the matching weight for the page in the reference ontology multiplied by the mapping factor for the reference ontology concept to the personal concept.

$$new\ weight = matching\ weight\ for\ page\ in\ reference\ ontology * mapping\ factor$$

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

After all pages have been mapped and their weights recalculated, the weights must be accumulated for the tree. The process of accumulating the weights is accomplished in the same manner as the LCA. Now, an LCA-mapped site can have its content browsed by the LBA or RBA using the personal ontology rather than OBIWAN's reference ontology.

6. 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 with the LBA, after it had been mapped to their personal ontology, see figure 4. Each user would decide if each page that was mapped to their personal ontology was mapped to the correct concept.

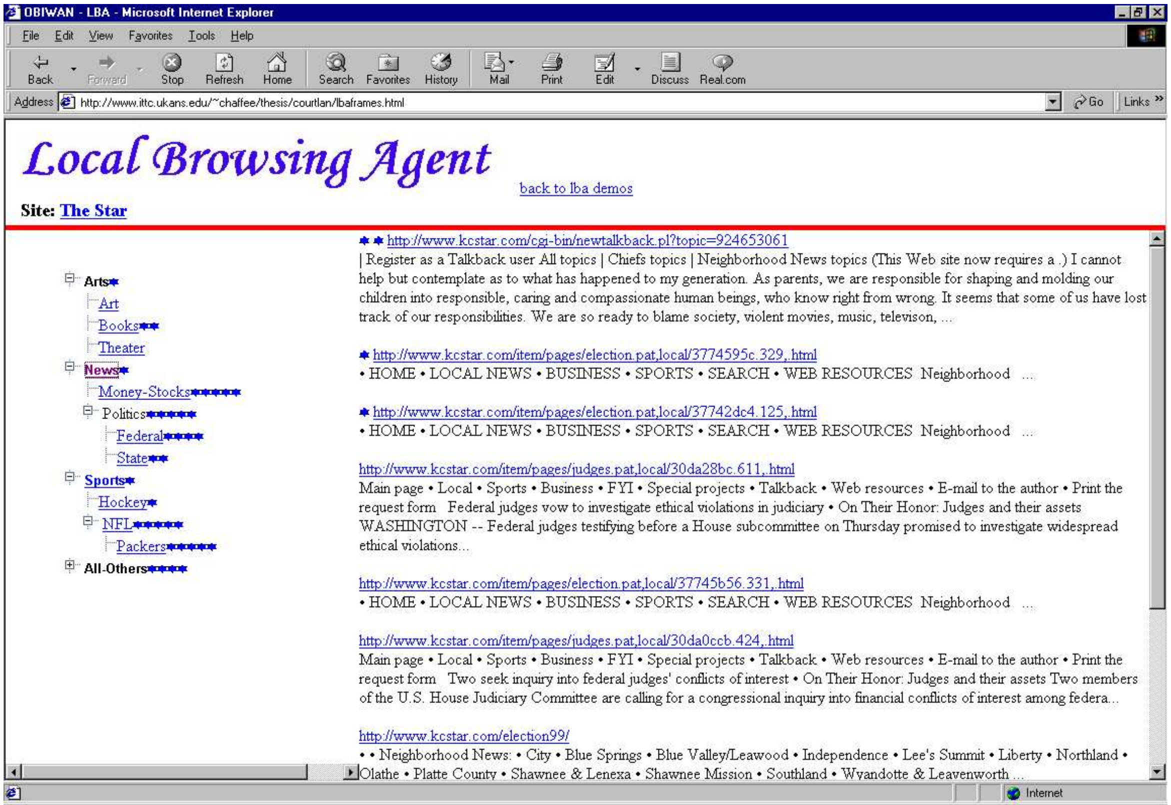

Figure 4. Screen shot of the LBA displaying the personal ontology.

6.1 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. Table I shows the precision, recall and correctness values for each 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. It was found that a threshold of 0.3 produces the most c orrect mappings.

$$correctness = \frac{number\ kept\ that\ are\ correct + number\ dropped\ that\ are\ incorrect}{total\ number\ of\ concepts\ mapped\ with\ no\ threshold}$$

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

Table I. Precision, recall and correctness values for concept mapping thresholds.

6.2 Evaluating Site Mappings

The evaluation of the ontology mappings showed that a threshold of 0.3 would produce the most correct mappings from the reference ontology to the personal ontology. 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. 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. Table II shows the precision, recall and correctness values for the page mappings at each threshold.

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

Table II Precision, recall and correctness values for page mapping thresholds.

6.3 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.

6.3.1 Concept Mappings

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 weren't always conceptual in nature, e.g., a user's name. Second, the training data in both the reference ontology and the personal ontologies was less than adequate. Either the pages contained very little content, or the content they contained had a template around it that added noise to the frequency stats of words in the template and also the concept vector for the page inside the template.

6.3.2 *Page Mappings*

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 LCA mapped the Web site pages. If the mappings of the concepts were correct, but the mapping of the individual Web pages into the reference ontology were incorrect, then they would be incorrect in the personal ontology.

7. CONCLUSIONS AND FUTURE WORK

This paper presented the idea of using personal ontologies to browse the Web. This was accomplished by mapping from a reference ontology to a personal ontology. Then, each user was allowed to browse a particular site with their personal ontology.

This paper showed that is possible to map between two ontologies. It was found that a precision and a correctness of 49% were possible with no threshold. When a threshold of 0.3 was used for the mapping factor we were able to obtain a precision of 53% and a correctness of 55%. Also, this paper showed that it was possible to map individual Web pages based on the mappings between two ontologies. It was found that a precision and a correctness of 50% were possible with no threshold. A threshold was not found to help these results because the pages were classified incorrectly into the reference ontology.

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. The reference ontology's training data is gathered by a spider that decides whether to collect the document or not based on the number of bytes. The training data would be improved if only content pages (those containing text rather than links and/or images) were identified. Users provide the personal ontology's training data. This causes the same problems that occur with gathering the reference ontology's training data. A "smarter" spider could be used to collect the training data for the personal ontology as well as the reference ontology.

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.

11. REFERENCES

[1] Lawrence, Steve and Giles, Lee. "Accessibility and Distribution of Information on the Web", http://www.wwwmetrics.com, 1999.

[2] Search Engine Watch. "*Search Engine Coverage Sizes*", http://searchenginewatch.com/reports/sizes.html, 2001.

[3] Cassola, E. ProFusion PersonalAssistant: An Agent for Personalize Information Filetering on the WWW. Master's thesis, The University of Kansas, Lawrence, KS, 1998.

[4] Zhu, X., Gauch, S., Gerhard, L., Kral, N. and Pretschner, A. Ontology-Based Web Site Mapping for Information Exploration, *Proc. Of the Eighth International Conference on Information and Knowledge Management* (*CIKM '99)*, Kansas City, MO, November 1999, 188-194.

[5] Lycos. "Lycos: Your Personal Internet Guide", http://www.lycos.com , 2002.

[6] Frakes, William B. and Baeza-Yates, Ricardo. *Information Retrieval Data Structures & Algorithms.* Prentice Hall, 1992.

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

[8] Armstrong, R., Freitag, D., Joachims, T., and Mitchell, T. Web Watcher: A Learning Apprentice for the World Wide Web. In *Proc. AAAI Spring Symp. on Information Gathering from Heterogeneous, Distributed Environments,* March 1995.

[9] Joachims, T., Freitag, D., Mitchell, T. Web Watcher: A Tour Guide for the World Wide Web. In *Proc. IJCAI'97*, August 1997.

[10] Mladenić, D. Personal Web Watcher: Design and Implementation. Technical Report IJS-DP-7472, J. Stefan Institute, Department for Intelligent Systems, Ljubljana, Slovenia, 1998.

[11] Chen, L. and Sycara, K. A Personal Agent for Browsing and Searching. In *Proc. 2nd Intl. Conf. on Autonomous Agents*, pp. 132-139, New York, USA, 1998

[12] Moukas, A. Amalthaea: Information Discovery and Filtering Using a Multiagent Evolving ecosystem. In *Proc. 1st Intl. Conf. on the Practical Application of Intelligent Agents and Multi Agent Technology*, London, 1996.

[13] Pazzani, M. Muramatsu, J. Billsus, D. Syskill & Webert: Identifying Interesting Web Sites. In *Proc. 19th Natl. Conf. on Artificial Intelligence*, 1996.

[14] Widyantoro, D., Ioerger, T. Yen, J. An Adaptive Algorithm for Learning Changes in User Interests. In *Proc. 8th Intl. Conf. on Information and Knowledge Management*, pp. 405-412, 1999.

[15] Liebermann, H. Letizia: An Agent that Assists Web Browsing. In *Proc. Intl. Conf. on AI*, August 1995.

[16] Liebermann, H. Autonomous Interface Agents. In *Proc. ACM Conf. on Computers and Human Interaction (CHI'97)*, Atlanta, USA, May 1997.

[17] Hsu, Wen-Lin and Lang, Sheau-Dong. Classification Algorithms for NETNEWS Articles. In *Proc. 8th Intl. Conf. on Information and Knowledge Management*, pp. 114-121, 1999.

[18] Göver, N., Lalmas, M., and Fuhr, N. A Probabilistic Description-Oriented Approach for Categorising Web Documents. In *Proc. 8th Intl. Conf. on Information and Knowledge Management*, pp. 475-482, 1999.

[19] Matsuda, K. and Fukushima, T. Task-Oriented World Wide Web Retrieval by Document Type Classification. In *Proc. 8th Intl. Conf. on Information and Knowledge Management*, pp. 109-113, 1999.

[20] Larkey, L. Automatic Essay Grading Using Text Categorization Techniques. In *Proc. 21st Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Melbourne, Australia, 1998.

[21] What is an Ontology. http://www.-ksl.stanford.edu/kst/what-is-an-ontology.html, 1999.

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

[23] Knight, K. and Luk, S. Building a Large Knowledge Base for Machine Translation. In *Proc. Amer. Assoc. Artificial Intelligenc Conf. (AAAI)*, pp. 773-778, 1999.

[24] Labrou, Y. and Finin T. Yahoo! As an Ontology – Using Yahoo! Categories to Describe Documents. In *Proc. 8th Intl. Conf. on Information and Knowledge Management*, pp. 180-187, 1999.

[25] Yahoo!, http://www.yahoo.com, 2000.

[26] Chowder, G. and Nicholas, C. Resource Selection in Café: an Architecture for Networked Information retrieval. In *Proc. SIGIR'96 Workshop on Networked Information Retrieval*, Zurich, 1996.

[27] Chower, G. and Nicholas, C. Meta-Data for Distributed Text Retrieval. In *Proc. First IEEE Metadata Conference*, 1996.

[28] Pearce, C. and Miller, E. The Telltale Dynamic Hypertext Environment: Approaches to Scalability. In *Advances in Intelligent Hypertext*, Springer-Verlag, 1997.

[29] Heflin, J., Hendler, J. and Luke, S. SHOE: A Knowledge Representation Language for internet Applications. Technical Paper, Institute for Advanced Computer Studies, University of Maryland, College Park.

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

[31] Pretschner, A., Gauch, S. Ontology Based Personal Search. In Proc. 11th IEEE Intl. Conf. on Tools with Artificial Intelligence, pp. 391-398, Chicago, November 1999.

[32] Yang, Yiming and Liu, Xin A Re-examination of Text Categorization Methods. In *Proc. 8th Intl. Conf. on Information and Knowledge Management*, pp. 42-49, 1999.

[33] Singhal, A., Buckley, C., Mitra, M., and Salton, G. Pivoted Document Length Normalization. *In Proc. 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 21-29, 1996.

---

* 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.