Ontology-Based Web Site Mapping for Information Exploration[1]

Xiaolan Zhu
Susan Gauch
Lutz Gerhard
Nicholas Kral
Alexander Pretschner

Department of Electrical Engineering and Computer Science
University of Kansas

Contact Information
Dr. Susan Gauch
415 Snow Hall
Dept. of EECS
University of Kansas
Lawrence, KS 66049
785-864-7755 (phone)
785-864-7789 (fax)
sgauch@eecs.ukans.edu

Abstract

A centralized search process requires that the whole collection reside at a single site. This imposes a burden on both the system storage of the site and the network traffic near the site. This argues that the search process should be distributed. Recently, more and more Web sites provide the ability to search their local collection of Web pages. Query brokering systems are used to direct queries to the promising sites and merge the results from these sites. Creation of meta-information of the sites plays an important role in such systems - it is used by the query brokers to identify appropriate sites to send each query. In this article, we introduce an ontology-based web site mapping method used to produce conceptual meta-information based. We present a series of experiments comparing our classification approach with Naive-Bayes approach, finding it more accurate. We also show how the automatically generated site map is used in distributed search, browsing and visualization in our multi-agent system, Obiwan.

Categories: distributed collections, information brokers, text categorization, IR agents (general).

---

[1] This project is partially supported by the National Science Foundation CAREER Award 97-03307.

Note: The primary author is a full time student and will present the paper if it is accepted. Thus, this paper is eligible for the Best Student Paper award.

1 Introduction

The World Wide Web (WWW) offers the promise of unlimited access to electronic information. Unfortunately, the reality is electronic access to unlimited anarchy. When information was first made available on the WWW, hundreds of thousands, then millions, of sites placed archives of information online. The information was known only to savvy users who, through serendipity, stumbled across it. To bring order to chaos, *spiders* were created to surf the Web and collect Web pages in a central location for indexing. Although the Web pages themselves were still distributed, the result was a handful of overtaxed computers that processed the queries of millions of users a day. Searching the Web went from a highly distributed activity to one that was completely centralized.

Recently, more and more Web sites are providing the ability to search their local collection of Web pages. Query brokering systems make use of this ability by sending queries to selected local Web sites for processing. To be a viable alternative to centralized search engines, several issues must be resolved, particularly scalability, how to identify relevant sites and how to fuse information from multiple sites. In this project, we focus on the automatic creation of conceptual meta-information for local sites, specifically weighted ontologies, which are used by regional information agents.

Browsing is another method to find information. Several search engines provide subject hierarchies that can be browsed, but the associated Web pages are manually placed in the categories, which limits the amount of information available. In contrast, we build a browsing structure automatically, by producing weighted ontologies via categorizing documents on the local site. In addition, just as the shared meta-information allows the user to search multiple sites simultaneously via a query broker, our shared meta-information can also be used to allow users to browse multiple sites simultaneously. We are exploring the use of visualization of the meta-information for individual and multiple sites as a mechanism to provide users with an overview of the information space.

The meta-information created for the local sites is the foundation on which the rest of our project rests. Thus, after providing an overview of our project and related work, this paper will focus on the experiments we have run to assess the accuracy of our automatically generated meta-information. Finally, we will conclude with the initial versions of our searching, browsing and visualization agents.

2 System Overview

We employ distributed, intelligent agents to organize information on the Web. As shown in Figure 1, each participating Web site has local agents that characterize and provide access to the information at the local site. These local agents in turn communicate with regional agents that characterize and provide access to the information for regions of the Web.

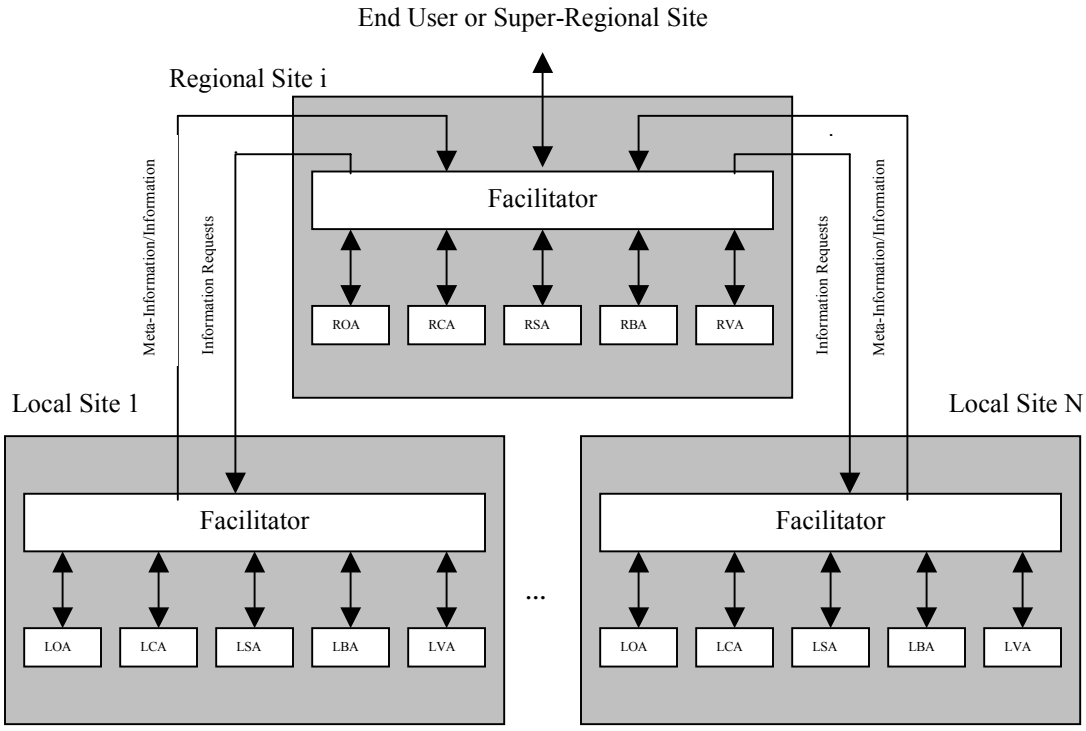

Figure 1: System Architecture for the Local and Regional Agents

Characterizing a Web Site The *Local Ontology Agent* (LOA) produces an ontology which represents the concepts contained in the local Web site.[2] The *Local Characterizing Agent* (LCA) takes this local ontology and the site's Web pages and produces a characterization of the local site with two components: a searchable index and a *site map* which is an association of words, pages, and evidence weights with each concept in the local ontology.

Accessing Information about a Web Site The searchable index is used by the *Local Search Agent* (LSA) to produce the urls of pages relevant to a particular query. The local site map is used by the *Local Browsing Agent* (LBA) to allow concept-based browsing of the site's contents. The local site map's evidence weights are used by the *Local Visualization Agent* (LVA) to provide a graphical representation of the site as a whole at different levels of abstraction. Finally, the historical information about the creation and/or destruction of new concepts in the ontology and the addition and/or subtraction of words associated with each concept are used to monitor the growth and adaptation of the contents of the local site.

Characterizing a Region The local sites co-operate to share their local site maps to provide a characterization of the World Wide Web. Logically, each site has a *Facilitator Agent* which communicates with the outside world. The Facilitator Agent provides the local ontology to a *Regional Ontology Agent* (ROA) that merges the ontologies from

[2] Currently, all local characterizing agents use the same local ontology.

multiple Facilitators to produce the *regional ontology*. Similarly, multiple Facilitators each provide their local site maps which in turn are merged to produce the *regional site map* by the *Regional Characterizing Agent* (RCA). Like the local concept bases, the regional concept base associates words, pages, and evidence weights with each concept in the regional ontology. However, in addition, promising local Web sites are identified for each concept in the regional ontology. In multi-level agent topologies, regions themselves have Facilitators that communicate with super-regions and so on, but their function is the same as described here.

Accessing Information about a Region Through the Facilitator, users may choose to search, browse or visualize a region of the Web rather than the local site. When a user wishes to search a region, the Facilitator directs the query to the *Regional Search Agent* (RSA). Based upon the contents of the query, the Regional Search Agent identifies the most applicable concepts in the regional concept base and then maps from those concepts to the most promising Web sites. The query will then be forwarded to those sites for processing by the appropriate Local Search Agents. Finally, the Regional Search Agent merges the results and returns them to the Facilitator for presentation to the User. Similar to the Local Browsing Agent, the regional concept base can be used by the *Regional Browsing Agent* (RBA) to allow concept-based browsing. The best Web pages associated with each concept could be displayed or the best Web sites or a mix of both. When a user comes upon a promising site, selecting that site should transfer control to the appropriate Local Browsing Agent. The interaction of the *Regional Visualization Agent* (RVA) with the Local Visualization Agents is exactly analogous.

3 Related Work

The Local Characterizing Agent produced the weighted site map by using a text categorization technique to map the Web pages in a local site to concepts in the ontology. This site map is then used by other agents for searching, browsing, and visualization in a distributed environment. In this section, we briefly overview the techniques used in site mapping, text categorization, distributed search, and agent communication.

3.1 Site Mapping

Most information services can be categorized into two types: search-based and browsing-based. The browsing-based information services provide users with an overview of a site to help them locate the Web pages of their interest via navigation. The browsing-based information services thus require a tool that creates a site map and a method to display the site map.

3.1.1 Site Map Generation

A site map represents the information space of a site by either relating a group of Web pages to a particular subject or depicting the links specified in the Web pages. To generate a *subject-based site map*, one must describe a subject space, which typically has a hierarchical structure, and then associate documents with the appropriate subjects.

Manual approaches, for example Infoseek's Ultraseek (Infoseek, 1998), concentrate on providing the user with tools which help them create topics and assign documents to the appropriate categories based on rules. Another approach is to analyze the contents of the Web pages in a site to produce a set of statistical data and indices, which are then used to train a neural network. The outcome of the neural network is a representation of a site map that contains labeled subject areas displayed with the urls of their related documents (Lin, 1995). Yet another approach to subject-based site map generation, used by a commercial product *ThemeScape* (Cartia, 1998), clusters the documents collected from a site and identifies the cluster topics and topic relationships based on the contents of the documents.

A *link-based site* map is usually generated by a spider that starts off from a set of initial urls to collect the Web pages in the site. A site map produced this way, for example *WebCutter* (Maarek et al., 1997), represents the organization of a set of urls and their interconnections.

3.1.2 Site Map Display

Site map display techniques attempt to present users with a clear global view of the site while, generally, allowing them to quickly navigate to areas of interest within the site. Simple spatial arrangement of subject titles may be used to group related information together. More sophisticated graphical techniques using shape, color, and three-dimensional views may also be used. Common examples are Fisheye Views (Furnas, 1986), Cone Trees (Robertson *et al.*, 1991), Treemaps (Johnson & Shneiderman, 1991), and Value Bars (Chimera, 1992). We are developing with simple text-based displays for browsing tasks and three-dimensional techniques for visualization tasks.

3.2 Text Categorization

Since our site map generation technique uses of text categorization, we will review C4.5 and Naïve-Bayes, which have been found to be among the most effective classifiers (Friedman & Goldszmidt, 1996).

In the C4.5 approach (Quinlan, 1993), the pre-defined categories are described in terms of a fixed collection of attributes. The descriptions of the categories are obtained by using an inductive generalization process that identifies the patterns and the attributes in a set of training data to construct classifiers, which can be represented as decision trees or sets of production rules. Each category therefore corresponds to a logical expression that specifies the values of the attributes. When new documents come in, it extracts the values of the attributes from them and then classifies them into particular categories based on the established decision tree or rules.

In the Naïve-Bayes approach (Langley et al, 1992), however, the conditional probability of each attribute value given a particular category is determined as well as the probability of the category appearing. This probabilistic information is typically established through a learning procedure, which takes in a new training instance each time and adjusts the probabilistic information of the corresponding category. For a new test instance, it evaluates and ranks the candidate categories, and then classifies the

instance to the top-ranked category based on the probabilistic information of the categories. This approach assumes that the attributes are probabilistically independent.

3.3 Distributed Search

The Internet has been witnessing the proliferation of search engines that have been developed to help users find information on the Web. However, each search engine indexes only a subset of the Web. To address this, meta-search engines have been developed which select a set of promising search engines for each query, submit the request simultaneously to the selected search engines, fuse and present the results to the users. Distributed search systems are similar to meta-search engines, but they may send queries to the document collections themselves, rather than to centralized search engines. Distributed search is divided into three steps: choose the best sites for a particular query, query the selected sites, and merge the search results from these sites.

3.3.1 Query Routing

The task of query routing is to locate the best sites for a given query in a distributed search environment. Query routing algorithms usually rely on meta-information which describes the contents and capabilities of each site. Different query routers require different types of meta-information. Content summaries (Gravano *et al.*, 1997), which contain a list of words that appear in the collection, the frequency of each word in the collection, and the total number of documents that contain a particular word, can be used as meta-information. N-gram centroids have also been used as meta-information (Crowder & Nicholas, 1996a; 1996b). The query router matches the n-gram profile of a given query to the n-gram centroids of sites and routes the query to the sites that have the best match. Term similarity matrices have also been used as meta-information for query routing and found more effective than content summaries (Gauch, Wang & Rachakonda, 1998). For each document collection, term-term similarities are calculated using corpus analysis techniques. The query router uses the similarity matrices to choose those collections which contain the richest set of terms similar to those used in the query.

3.3.2 Information Fusion

In a distributed search environment, the rankings of the results rely on the individual collections' overall relevance to the query. The *N* best documents from a site with little information relevant to a query are not comparable with the *N* best documents from a different collection that contains more relevant information. Overall retrieval effectiveness can be severely degraded if the responses from different collections are simply ordered by the rankings reported by the individual sites (Voorhees, 1997). One approach to information fusion attempts to maximize the total number of relevant documents retrieved by requesting more documents from those collections believed to contain more relevant information (Towell *et al*., 1995; Voorhees, 1997; Callan, Croft & Harding, 1995). In one approach (Gauch, 1997; Fan & Gauch, 1999), rankings are

reassigned based on two factors: the value of the query-document match reported by a site, *the match factor*, and the estimated accuracy of the site, *the confidence factor*.

3.3.3 Agent Communication

Agent-based information systems typically require a group of agents working together to fulfil a common goal. There are a number of different agent architectures that allow agents to coordinate and cooperate with each other (Genesereth & Ketchpel, 1994). In the direct communication approach, agents talk to each other directly using either *contract nets* or *specification sharing* whereas in a *federated system* agents are organized in a hierarchy. Federated agents communicate only with a facilitator that coordinates activity among the agents in the federation (Weiss *et al.*, 1996; Marian *et al.*, 1998). There are two major approaches used by agent communication languages/protocols (Cheong, 1996). *Declarative approaches*, such as KQML (Neches *et al*., 1991) and CORBA (OMG, 1998), are commonly used to implement platform-independent information agents whereas *procedural approaches*, such as Java, Tcl, Safe-Tcl, Smalltalk, Perl, Python, Telescript, and ActiveX, are usually used for mobile agents (Caglayan & Harrison, 1997).

4 Generating Site Maps

The Local Characterizing Agent (LCA) creates a site map that consists of a hierarchy of subjects and associated weights where the weights indicate the amount of the information relevant to that subject at the local site. Since the other agents are primarily based on the data produced by the LCA, it plays a central role in the agent community. The LOA decides if a subject should be removed or added to the local ontology based on the data provided by the LCA that gives information about if there are some Web pages relevant to the subject on the site. The LBA and the LVA both create a representation of the global view of the local site based on the site map generated by the LCA. The LSA searches the local site based on the index produced by the LCA. The site map it produces is also used as meta-information by the regional agents (see Section 5).

4.1 Approach

The LCA is given a pre-existing ontology (in our case, a browsing hierarchy used by a prominent Web site) and categorizes each Web page from the local site into the best matching concept. The documents that have been manually attached to each concept in the browsing structure are used as training data for categorizing the new documents. We use a vector space approach, calculating the cosine similarity measure to identify the closest match between a vector representing the Web page and the vectors representing the concepts in the ontology. All the documents associated with a given concept are concatenated to form a *superdocument*. These superdocuments are then indexed using a vector space retrieval engine (the same one used by the LSA). Each Web page at the local site is then treated as a query and the retrieval engine returns the top matching superdocuments for that "query" (i.e., the top matching categories for that Web page).

Each document is attached to only the top 10 matching concepts in the ontology. The weight of the document that matches to the concepts is then accumulated, which in turn contribute to the overall concept weights of their ancestors in the ontology.

To validate the quality of our categorization, the next section discusses series of experiments that compare the performance of the vector space categorization approach with one of the top classifiers, Naïve-Bayes (Friedman *et al*., 1997).

4.2 Experiments

We compared the performance of our approach with a Naïve-Bayes implementation from Carnegie-Mellon University (McCallum, 1998) and found our method was more accurate.

Method We selected an ontology containing 1,274 concepts and spidered 10 associated Web pages for each concept. We evaluated the accuracy of classifiers with a variable numbers of Web pages (one through eight per concept) in the training set, using the remaining two Web pages per concept as the test set.

Results and discussion Table 1 shows the mean distance between the concept chosen by the classifiers and the concept with which the test document was originally associated. We defined the distance between two concepts as the length of the path between them. For example, the distance between a parent and its child in the ontology would be one and the distance between siblings would be two. Since the ontology had a maximum depth of 11, the maximum possible distance between the assigned and target concepts would be 22. Table 1 also shows the number of exact matches, where the classifier chose the same category with which the Web page was originally associated. The results in Table 1 are based on classifying two test documents per concept (i.e., 2,548 test documents) when variable numbers of pages per concept were used for training. The statistical t-test analysis shows that the mean distance produced by the vector space approach is significantly smaller than that for the Naïve-Bayes approach ($t(14) = 5.56$, $p < .001$). The number of matches for the vector space is also significantly larger than that for the Naïve-Bayes approach ($t(14) = 4.45$, $p = .001$).

Training Pages per Concept	**Vector Space**		**Naïve-Bayes**	
	Mean Distance	**Number of Matches**	**Mean Distance**	**Number of Matches**
1	6.0687	127	7.0667	32
2	5.6943	202.5	6.8697	61
3	5.1091	289	6.6138	95.5
4	4.6920	355	6.4270	127
5	4.4157	407.5	6.2901	149
6	4.2657	434.5	6.1084	184
7	4.3199	423.5	6.0887	188
8	4.3882	415	5.9620	184
Average	**4.8692**	**331.75**	**6.4283**	**127.56**

Table 1: Distance and Number of Matches for the Vector Space and Naïve-Bayes Approaches

Examining Table 1 in more detail, we can see the effect of the amount of training data on the classification accuracy, which shows that, for the Vector Space method, the highest accuracy occurs with six training pages per concept. Adding more pages per concept degrades it slightly.

Figure 2 illustrates the classifiers' performance in more detail, showing the histogram of distances when the number of training documents was six.

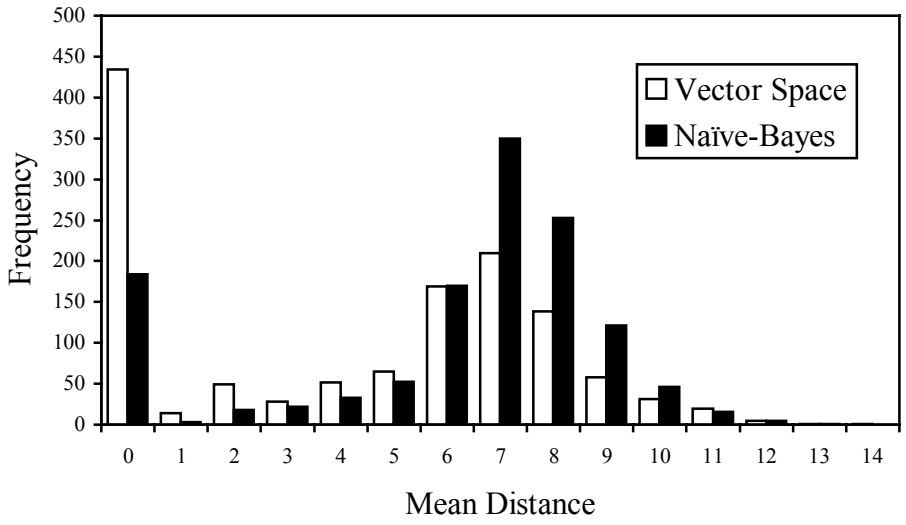

Figure 2: The Histogram of Distances between Assigned Concept and the Correct Concept (Number of Training Pages = 6) for the Vector Space Approach and the Naïve-Bayes Approach.

In addition to varying the number of training pages, we also investigated the effect of using only a subset of words rather than the complete test documents as the input to the vector space classifier. The words were weighted using tf * idf (where tf was the frequency of the term in the test document and idf was the inverse document frequency calculated over all available training documents). Figure 3 shows the results when we varied the number of words selected. The data show that the mean distance decreased as the number of selected words increased, but little improvement is seen beyond 40 words selected.

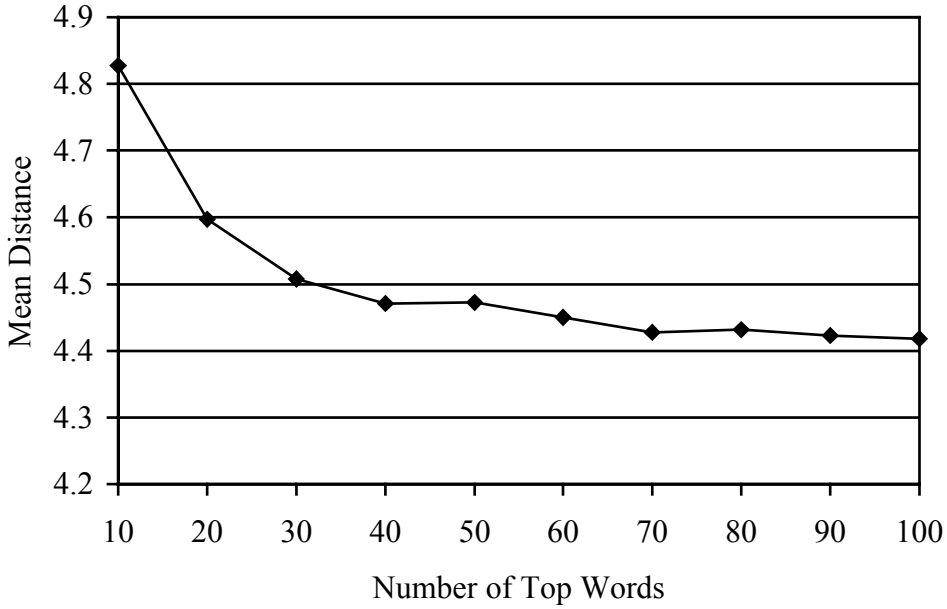

Figure 3: Mean Distance as a Function of the Number of Top Words in Test Documents

In summary, the data from the experiments show that the vector space approach was more accurate than the Naïve-Bayes approach, and that a modest number of training documents (approximately six per concept) are sufficient for good performance. The results also show that a small number of the most important words for a document convey enough information for classification. Overall, the site map generated by the LCA is seems to be quite accurate for an entirely automatic approach.

5 The Site Map as Meta-Information

The Regional Characterizing Agent (RCA) receives meta-information (the site maps) from the local sites in the region and merges the local site maps into a regional site map. A regional site map contains the weight representing the *quantity* of pertinent information the Web sites in the region collectively contain for each concept in the regional ontology. Through studying the actions of the users of the information (do they bookmark the pages, or print them, or at least read them) the Regional Browsing, Search, and Visualization Agents learn the *quality* of the information provided by the various local agents. Based on this quality information, the RCA assigns reliability measures to the local sites, which is used when constructing the regional knowledge base for query routing and information fusion.

5.1 Search

The tasks of the Regional Search Agent (RSA) include directing queries to the best sites (based on quantity and quality) and fusing the results from the multiple responses. Using the same training data used by the LCA, incoming queries are matched against concepts in the ontology. For each of the top ten matching categories, the quantity of information each of the local sites in the region contains is determined from the meta-information they provide. These quantities are accumulated for each site, and the query is brokered to the top matching sites for local processing. We reassign the match values for the results from the different sites based on two factors: the value of the query-document match reported by a site, *the match factor*, and a function of the quantity and quality of information at the site, *the confidence factor* (Fan & Gauch, 1999). Figure 4 shows an example when site i and site j are selected by the RSA. Although the communication between the RSA and the LSAs is performed using sockets for efficiency, the LSAs can also communicate with agents from other projects using CORBA. The RSA spawns a thread for each request, which then handles site selection and information fusion for this particular request.

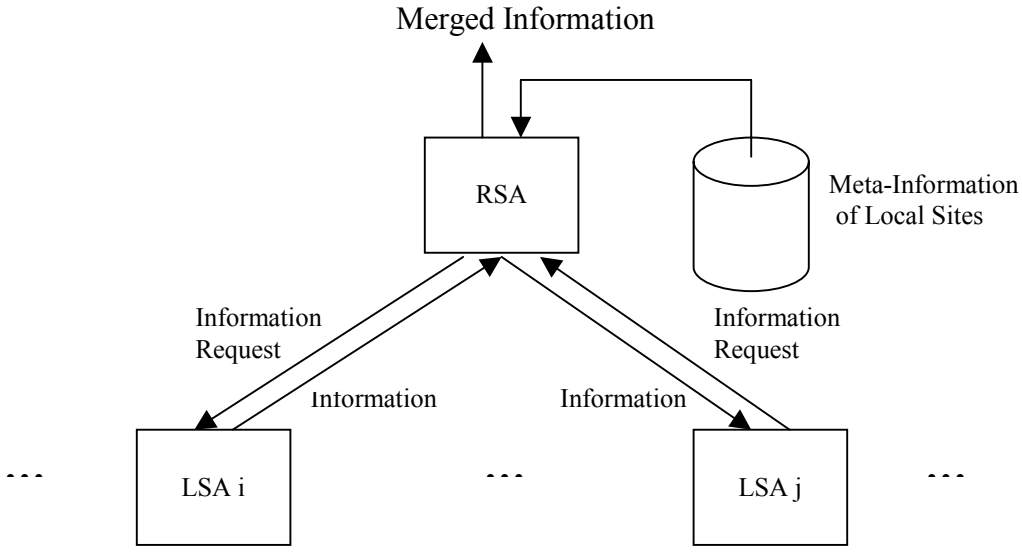

Figure 4: Query Brokering using Conceptual Meta-Information

5.2 Browsing

The goal of the LBA (Figure 5) is to guide users to reach the Web pages of their interest. It provides the user with the site map in a textual hierarchy which is implemented in Java for smooth user interaction. The user can locate the Web pages by following the relevant subjects in the hierarchy. The RBA (Figure 6) is different from the LBA in that it provides the regional site map, accumulated from the local site maps, to users. Rather than directing users to Web *pages* of interest, it guides the user to the most promising *sites* that contain information pertinent to their interest. The RBA is also implemented as a Java applet, and the user can move seamlessly from the Regional to the Local Browsing agent.

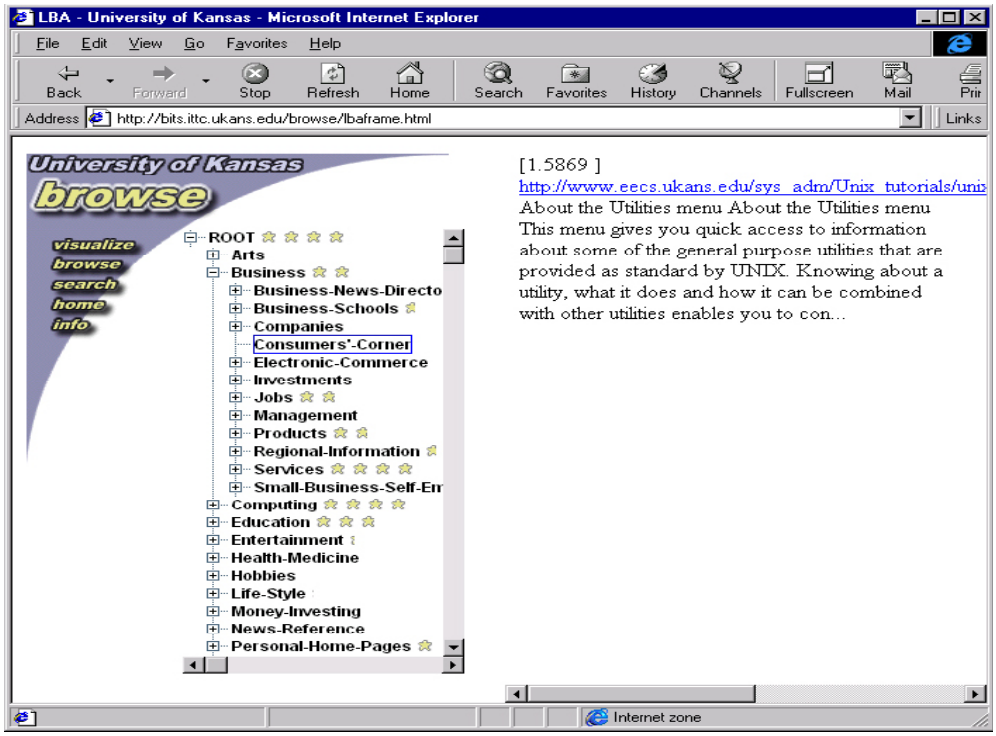

Figure 5: A Screen Shot of the Local Browsing Agent (LBA).

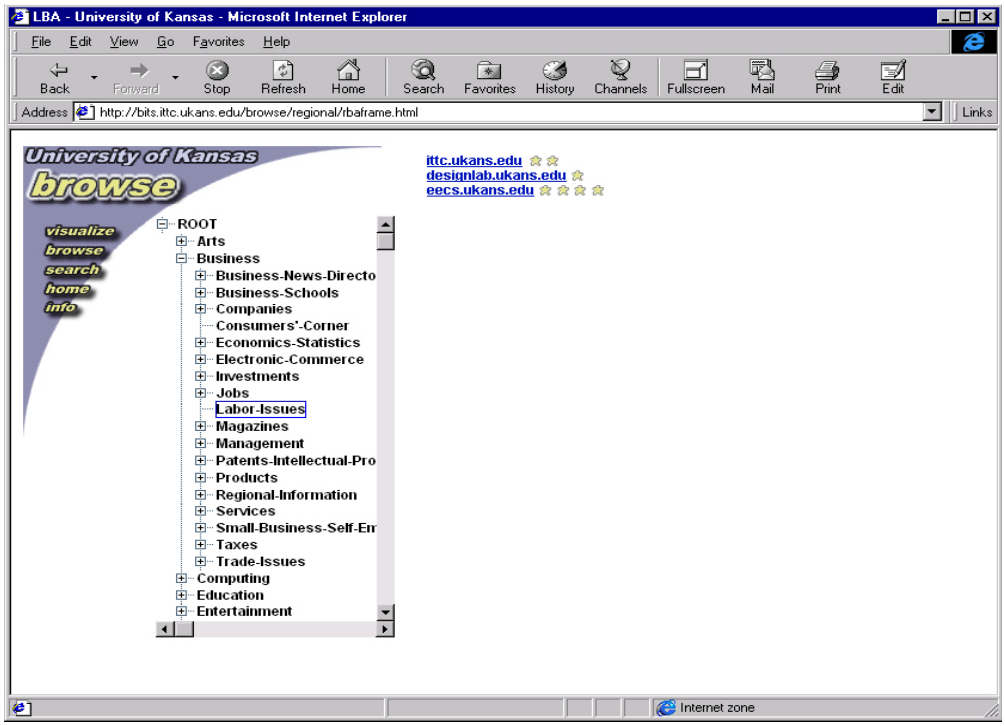

Figure 6: A Screen Shot of the Regional Browsing Agent (RBA).

5.3 Visualization

The Local and Regional Visualization Agents will present the local and regional site maps, respectively, using three-dimensional visualization techniques. The goal of these agents is to convey the most important parts of the local and regional site maps at a glance. We have a preliminary implementation of the LVA, but it is not described further here since it will likely change as we develop the RVA.

6. Conclusions

We have described an agent-based distributed information exploration system. The LCA plays a central role in the system. It produces the conceptual meta-information for a local site, which is used as a guideline for distributed searching, browsing and visualization. High quality meta-information therefore leads to a more useful information system. We introduced an ontology-based site mapping approach, which uses a vector space method for text classification. We presented experiments evaluating the effectiveness of our classification approach and found that the classification was more accurate than Naïve-Bayes. In particular, the number of pages assigned to the correct concept was significantly higher, as was the mean distance between the assigned concept and the correct concept. We found that six Web pages are sufficient for training our classifier and that the top weighted 40 words from a Web page were sufficient for accurate classification.

Our system combines searching, browsing, and visualization abilities in a integrated distributed information exploring system. We have completed regional agents that use the meta-information from the local sites for search and browsing, but these approaches need to be formally evaluated. In addition, we are currently investigating the techniques to best visualize the conceptual meta-information for multiple sites. Future goals are to explore methods to allow the local ontologies to adapt to the contents of the individual Web sites, requiring the Regional agents to merge and manipulate disparate ontologies.

7 References

Caglayan, A. K. & Harrison, C. G. 1997. *Agent Sourcebook*. Wiley Computer Publishing.

Callan, J. P., Croft, W. B., & Harding, S. M. 1995. The INQUERY retrieval system. *Proceedings of the 3rd International Conference on Database and Expert System Applications*, September.

Cheong, F. 1996. *Internet Agents – Spiders, Wanders, Brokers, and Bots*. IN: New Riders.

Crowder, G. & Nicholas, C. 1996a. Using Statistical Properties of Text to Create Metadata. *First IEEE Metadata Conference*, April.

Crowder, G. & Nicholas, C. 1996b. Resource Selection in CAFÉ: an Architecture for Network Information Retrieval. *ACM-SIGIR96 Workshop on Networked Information Retrieval*, August.

Chimera, R., 1992. Value bars: An information visualization and navigation tool for multiattribute listings. *Proceedings. CHI'92 Conference: Human Factors in Computing Systems, ACM*, 293-294.

Fan, Y., & Gauch, S. 1999. Adaptive Agents for Information Gathering from Multiple, Distributed Information Sources. *Proceedings of 1999 AAAI Symposium on Intelligent Agents in Cyberspace*.

Friedman, N., Geiger, D., & Goldszmidt, M. 1997. Bayesian Network Classifiers. *Machine Learning*, 29, 131--163.

Friedman, N., & Goldszmidt, M. 1996. Building classifiers using Bayesian networks. *Thirteenth National Conference on Artificial Intelligence (AAAI)*.

Furnas, G.W. 1986, Generalized fisheye views. *Proceedings of CHI'86 Human Factors in Computing Systems, ACM*, 16-23.

Gauch, S. 1997. Information Fusion with ProFusion. *ACM-SIGIR97 – Workshop on Networked Information Retrieval*.

Gauch, S., Wang, J., & Rachakonda, S. M. 1998. A Corpus Analysis Approach for Automatic Query Expansion and its Extension to Multiple Databases. *ACM Transactions on Information Systems*.

Genesereth, M. & Ketchpel, S. 1994. Software Agents. *Commun. ACM*. 37(7), 48-53.

Gravano, L., Chang, C. C. K., Garcia-Molina, H., & Paepcke, A. (1997). STARTS: Stanford Proposal for Internet Meta-Searching. *Proceedings of the 1997 ACM*

SIGMOD International Conference on Management of Data.

Infoseek Home Page. 1998. http://software.infoseek.com/products/products.htm.

Johnson, B. and Shneiderman, B. 1991. Treemaps: A Space-filling Approach to the Visualization of Hierarchical Information. *Proceedings of IEEE Visualization '91 Conference*, 284-291.

Langley, P., Iba, W., & Thompson, K. 1992. An analysis of Bayesian classifiers. *Proceedings of the Tenth National Conference on Artificail Intelligence*. San Jose: AAAI Press.

Lin, X. 1995. Searching and browsing on map displays. *ASIS '95, Proceedings of the 58th ASIS Annual Meeting, Converging Technologies: Forging New Partnerships in Information*.

Marian, N., Perry, B., & Unruh, A. 1998. Experience with the InfoSleuth Agent Architecture. *Proceedings of AAAI-98 Workshop on Software Tools for Developing Agents.*

McCallum, A. 1998. A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. *http://www.cs.cmu.edu/~mccallum/bow/*

Maarek, Y., Ben-Shaul, I., Jacovi, M., Shtalhaim, M., Ur, S., & Zernik, D. 1997. WebCutter: A System for Dynamic and Tailorable Site Mapping. *The 6th International World Wide Web Conference*.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. 1991. Enabling technology for knowledge sharing. *AI Magazine*, 12(3), 36-56.

OMG Home Page. 1998. http://www.omg.org/.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. CA: Morgan Kaufmann.

Robertson, G. G., Mackinlay, J. D. and Card, S. 1991. Cone Trees: Animated 3D Visualizations of Hierarchical Information. *Proceedings of the ACM SIGCHI'91 Conference on Human Factors in Computing Systems*.

Salton, G., & McGill, M. 1983. *An Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Towell, G., Voorhees, E. M., Gupta, N. K., & Johnson-Laird B. 1995. Learning Collection Fusion Strategies for Information Retrieval. *Proceedings of the Twelth Annual Machine Learning Conference*.

Voorhees, E. M. 1997. Database Merging Strategies for Searching Public and Privated

Collections. *ACM-SIGIR97 Workshop on Networked Information Retrieval*.

Weiss, R., Velez, B., Sheldon, M. A., Namprempre, C., Szilagyi, P., Duda, A., & Gifford, D. K. 1996. HyPursuit: A Hierarchical network search engine that exploits content-link hypertext clustering. *Hypertext'96: The Seventh ACM Conference on Hypertext.*